

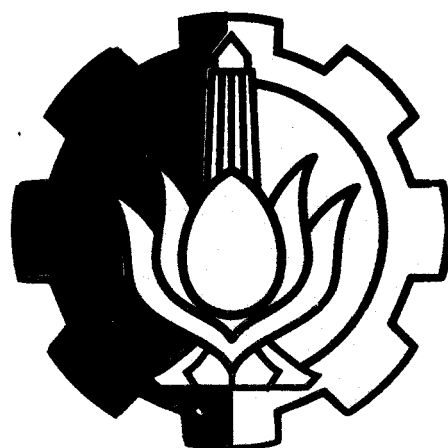
3100097008352

PENERAPAN METODA SHIFTING BOTTLENECK UNTUK PENJADWALAN JOB SHOP

TUGAS AKHIR

*Diajukan untuk Memenuhi Persyaratan Menyelesaikan
Studi Strata Satu dan Memperoleh Gelar Sarjana Teknik Industri*

RSI
658.53
Set
P-1
1996



Oleh :

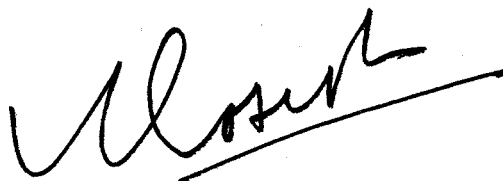
SETIONO

NRP : 2591 100 020

**JURUSAN TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1996**

PERPUSTAKAAN ITS	
Tgl. Terima	24 DEC 1996
Terima Dari	H
No. Agenda Prg.	6760

Mengetahui / Menyetujui
Dosen Pembimbing



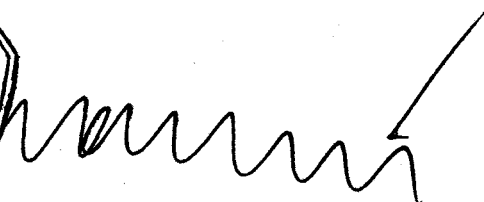
Ir. MOSES LAKSONO SINGGIH, M.Sc, MRegSc, Ph.D.

NIP. 131 694 604

Mengetahui,

Jurusan Teknik Industri
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
S u r a b a y a

Ketua,
Dekan FTI – ITS



SITOMO WIGNJOSOE BROTO, M.Sc.

NIP. 130 687 436

NILAI AKHIR SIDANG SARJANA TEKNIK INDUSTRI - ITS

Nama : Setiono

Nrp. : 2591.100.020

Komponen :

Komponen / Aspek (1)	Bobot (persen) (2)	Nilai Perolehan (3)	Nilai Angka (2) x (3)
1. Seminar (dari Seminar T.A)	15	83.5	12.5
2. Sidang (dari Sidang Sarjana)	45	82,3	37,0
3. Tugas Akhir (khusus diberikan oleh Dosen Pembimbing)	40	87	34,8
Jumlah	100		84,3

Nilai Huruf : (A) B C D E

Skala Nilai :

A = 81 - 100 B = 66 - 80 C = 56 - 65 D = 41 - 55 E = 0 - 40

Surabaya, 8 Okt 1996

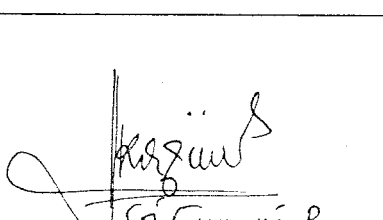
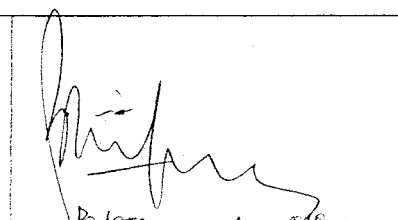
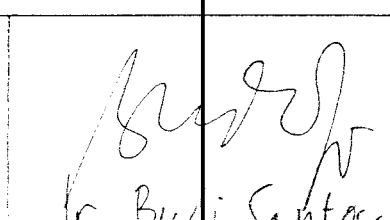

Setiono

Mahasiswa ybs.



Ir. Moses L. S., MSc. MReg.Sc. Ph.D.

Dosen Pembimbing

 Sri Gunani P. Dosen Penguji I	 BUSTANUL A. NOER Dosen Penguji II	 Ir. Budi Santosa Dosen Penguji III
---	---	---

*Kupersembahkan karyaku ini kepada
Ibu, Bapak, Kakak dan Adikku yang tercinta*

"Allah mengangkat orang yang beriman di antara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat".

(Surat Al Mudjadalah ayat 11)

..... Katakanlah: "Adakah sama orang-orang yang mengetahui dengan orang-orang yang tidak mengetahui?" Sesungguhnya orang yang berakallah yang dapat menerima pelajaran.

(Surat Az-Zumar ayat 9)

Exactly Righth For The First Time

*Berusahalah dengan benar sejak dari awal.
Memang kesempurnaan dalam bekerja mustahil kita raih, namun dengan selalu berusaha dengan benar sejak dari awal, Insyallah kita dapat mendekati kesempurnaan itu.*

ABSTRAKSI

Terdapat banyak metoda atau pendekatan yang dapat digunakan untuk membuat rencana penjadwalan produksi. Metoda-metoda atau pendekatan-pendekatan tersebut berbeda satu sama lain, baik masalah yang dipecahkan, cara yang digunakan dan solusi yang dihasilkan serta waktu yang diperlukan untuk memperoleh solusi. Penjadwalan dengan pendekatan optimal adalah perencanaan penjadwalan dengan suatu pendekatan yang memberikan solusi yang terbaik terhadap suatu masalah penjadwalan ditinjau dari kriteria tertentu, tetapi waktu dan kesulitan penyelesaian pendekatan optimal pada masalah yang relatif besar meningkat sesuai dengan tingkat kesulitan masalah. Oleh karena itu dikembangkan pendekatan heuristik yang biasanya memberikan solusi yang lebih cepat dari pendekatan optimal dengan solusi yang cukup memuaskan tetapi tidak menjamin hasil yang optimal. Penelitian terhadap masalah penjadwalan dengan pendekatan heuristik maupun dengan pendekatan optimal terus dilakukan untuk memperoleh solusi yang lebih baik dan atau memberikan waktu penyelesaian masalah yang lebih cepat.

Hampir semua prosedur heuristik penjadwalan job shop yang banyak dijelaskan dalam berbagai literatur merupakan penjadwalan dengan aturan "Priority Dispatching". Prosedur heuristik jenis ini menghasilkan solusi yang cepat dan biasanya memberikan solusi yang cukup baik. Dalam banyak situasi pendekatan ini sering dan banyak dibutuhkan dan cukup memadai untuk menyelesaikan masalah penjadwalan, tetapi dengan semakin meningkatnya kecepatan proses perhitungan oleh komputer dan semakin berkembangnya kebutuhan akan penjadwalan yang semakin efisien, maka semakin penting untuk menghasilkan jadwal yang lebih baik dan memberikan solusi yang mendekati optimal atau bahkan optimal dengan mengeluarkan tambahan biaya untuk proses komputasinya.

Shifting Bottleneck merupakan suatu metoda yang berbeda dengan teknik priority dispatching dalam membuat suatu jadwal. Dalam membuat jadwal, metoda ini menjadwalkan mesin-mesin satu per satu secara berturut-turut diantara mesin-mesin yang belum terjadwal yang bertujuan untuk menemukan suatu mesin yang bottleneck. Setiap suatu mesin bottleneck baru ditemukan, maka urutan atau jadwal untuk mesin-mesin yang telah terjadwal dilakukan reoptimasi lokal. Prosedur pencarian mesin yang bottleneck dan reoptimasi lokal keduanya didasarkan pada penyelesaian optimal penjadwalan satu mesin dengan menerapkan algoritma Schrage.

KATA PENGANTAR

Puji Syukur kami panjatkan atas kehadiran Allah SWT yang telah memberikan rahmat, hidayah dan karuniaNya sehingga penulis dapat menyelesaikan penyusunan dan penulisan laporan Tugas Akhir ini. Tugas Akhir ini dibuat untuk memenuhi persyaratan studi sarjana strata satu pada Jurusan Teknik Industri, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya. Tugas Akhir yang penulis susun disini berjudul “*Penerapan Metoda Shifting Bottleneck Untuk Penjadwalan Job Shop*”

Dalam penulisan Tugas Akhir ini banyak sekali hambatan yang ditemui, namun alhamdulillah semua itu dapat diatasi berkat bantuan dari banyak pihak baik bantuan berupa moril maupun materiil. Tanpa bantuan mereka sulit rasanya untuk dapat menyelesaikan Tugas Akhir ini dengan baik. Oleh karena itu pada kesempatan ini penulis ingin sekali mengucapkan terima kasih kepada :

1. Bapak Ir. Moses Laksono Singgih, M.Sc, MReg.Sc, PhD selaku dosen pembimbing yang telah banyak mencurahkan perhatiannya untuk membimbing, mengarahkan dan memberikan motivasi yang kuat dalam penyusunan Tugas Akhir ini.
2. Bapak Ir. Patdono Soewignjo, MEng.Sc. selaku Ketua Jurusan Teknik Industri ITS.
3. Ibu, bapak serta kakak dan adikku yang tercinta yang telah membantu memberikan bantuan baik moral maupun material.
4. Seluruh staff pengajar dari Teknik Industri yang telah memberikan ilmu serta wawasannya selama penulis mengikuti perkuliahan.

5. Ali Ridho yang telah membantu dalam pembuatan software job shop.
6. Setyo '92 dan Frans '93 yang telah merelakan printernya untuk dipinjam.
7. Teman-teman baikkku, Delta, Ulum, Santoso, Yani, Iwan, Baju, David, Togar dan juga masih banyak lagi teman lain terutama dari angkatan '91 yang selalu kompak dalam kuliah dan berdiskusi.
8. Nurrahman yang mau bertindak sebagai lawan diskusi berat dalam penyelesaian dan penyusunan Tugas Akhir ini.
9. Dan semua pihak yang tidak mungkin penulis sebutkan satu-persatu.

Akhir kata, semoga penulisan Tugas Akhir ini dapat berguna bagi semuanya meskipun dalam penulisannya masih jauh dari sempurna. Oleh karena itu kritik dan saran yang bersifat untuk menyempurnakan penulisan Tugas Akhir ini akan diterima dengan senang hati.

Surabaya, Oktober 1996

Penulis

DAFTAR ISI

BAB	halaman
LEMBAR JUDUL	i
LEMBAR PENGESAHAN	ii
LEMBAR PERSEMBAHAN	iii
ABSTRAKSI	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Perumusan Masalah	3
1.3. Tujuan dan Manfaat Pembahasan	4
1.4. Batasan Masalah	5
1.5. Sistematika Penulisan	6
II LANDASAN TEORI	
2.1. Studi Pengukuran dan Penetapan Waktu Kerja	8
2.1.1. Pengukuran Waktu Kerja Dengan Jam Henti	9
2.1.2. Penetapan Jumlah Siklus Kerja Yang Harus Diukur	12
2.1.3. Test Keseragaman Data	14
2.1.4. Uji Kesesuaian Distribusi Normal	15
2.1.5. Penetapan Waktu Baku	17
2.2. Definisi Penjadwalan Produksi	18
2.2.1. Penjadwalan Produksi	18
2.2.2. Terminologi Dalam Penjadwalan	23
2.2.3. Penjadwalan Job Shop	24
2.2.4. Jenis-jenis Jadwal	28
2.2.5. Teknik Priority Dispatching	31
III METODOLOGI PENELITIAN	
3.1. Tujuan Penelitian	33
3.2. Identifikasi dan Perumusan Masalah	33
3.3. Studi Literatur	34
3.4. Penelitian dan Pembuatan Model	34

	3.5. Pengumpulan dan Pengolahan Data	35
	3.6. Analisis Model	35
	3.7. Kesimpulan dan Saran	35
IV	PENYUSUNAN DAN PENGEMBANGAN METODA SHIFTING BOTTLENECK	
	4.1. Permasalahan	38
	4.2. Pendekatan Penyelesaian	41
	4.3. Algoritma Lintasan Terpanjang	46
	4.3.1. Algoritma Perhitungan Lintasan Terpanjang	47
	4.4. Prosedur Reoptimasi Lokal	48
	4.5. Penjadwalan Satu Mesin	50
	4.6. Pengembangan Model Dasar	58
	4.6.1. Waktu Siap Job Yang Tidak Sama Dengan Nol	58
	4.6.2. Waktu Siap Mesin Yang Tidak Sama Dengan Nol	60
	4.6.3. Adanya Job-Job Yang Berprioritas	61
	4.6.4. Adanya Routing Kompleks	62
V	PENGUMPULAN DAN PENGOLAHAN DATA	
	5.1. Aliran Proses Produksi	64
	5.2. Waktu baku Proses	76
VI	ANALISIS MODEL	
	6.1. Analisis Terhadap Model Dasar	84
	6.2. Analisis Perbandingan Dengan Model Lain	89
	6.3. Analisis Model Untuk Job Yang Tidak Siap Pada $t = 0$	90
	6.4. Analisis Model Untuk Mesin Yang Tidak Siap Pada $t = 0$	94
	6.5. Analisis Model Untuk Routing Job Yang Kompleks	97
	6.6. Analisis Model Untuk Job Berprioritas	100
	6.7. Analisis Model Untuk Penjadwalan Di PT. BBI Surabaya	103
VII	KESIMPULAN DAN SARAN	
	7.1. Kesimpulan	106
	7.2. Saran	107

DAFTAR PUSTAKA

LAMPIRAN A Uji Distribusi Normal

LAMPIRAN B Waktu Loading-Unloading dan Perhitungan Data

LAMPIRAN C Kasus-Kasus Penjadwalan Job Shop

LAMPIRAN D Listing Program

DAFTAR GAMBAR

GAMBAR	halaman
2.1. (a). Aliran Proses Flow Shop	
(b). Aliran Proses Job Shop.....	22
2.2. Aliran Kerja Pada Mesin Job Shop.....	25
2.3. Penggambaran Secara Grafik Problem Job Shop	
(a). Job by Job (b). Machine by Machine.....	26
2.4. Dua Bentuk Jadwal Job Shop	
(a). Gant Chart (b). Jadwal Berdasarkan Job.....	27
2.5. Penjadwalan Semiaktif Untuk $n = 2$ dan $m = 2$	29
2.6. Diagram Venn Hubungan Antara Jadwal Non Delay, Aktif dan Semiaktif	
(a). Jadwal optimal merupakan bagian jadwal non delay	
(b). Jadwal optimal diluar jadwal non delay	31
3.1. Langkah-Langkah	36
4.1. Disjunctive Graph 5 Job 4 Mesin	40
4.2. Flow Chart Algoritma Shifting Bottleneck	45
4.3. Flow Chart Reoptimasi Lokal	49
4.4. Flow Chart Algoritma Schrage	53
4.5. Pendefinisian J dan c	55
4.6. Pencabangan	57
4.7. Operasi 2 Siap Pada $t = t_s$	59
4.8. Mesin 2 Siap Pada $t = t_s$	60
4.9. Routing Kompleks	62
5.1. Intake Mazda 626	66
5.2. Intake Mazda 323	67
5.3. Exhaust Mazda 323	68
5.4. Exhaust Mazda MR 90	69
5.5. Crank Case Yanmar TS 230	70
5.6. Crank Case Yanmar TF 300.....	71
5.7. Counter weight.....	72
5.8. Cylinder Head	73
5.9. Bearing Cap.....	74
5.10. Trash Bearing	75
6.1. Disjunctive Graph Model Dasar	87
6.2. Gantt Chart Hasil Jadwal Model Dasar	88
6.3. Disjunctive Graph Job Tak Siap	91
6.4. Gantt Chart Hasil Jadwal Job Tak Siap	93

6.5.	Disjunctive Graph Mesin Tak Siap	95
6.6.	Gantt Chart Hasil Jadwal Mesin Tak Siap	97
6.7.	Disjunctive Graph Routing Kompleks	98
6.8.	Gantt Chart Hasil Jadwal Routing Kompleks	100
6.9.	Graph Job Berprioritas	101
6.10.	Graph Solusi Job Berprioritas	102

DAFTAR TABEL

TABEL	halaman
2.1. (a). Routing (b). Waktu Proses.....	27
2.2. Routing untuk Problem $n = 2$ dan $m = 2$	28
5.1. Waktu Loading-Unloading Intake Mazda 626.....	Lamp B
5.2. Waktu Loading-Unloading Intake Mazda 323	Lamp B
5.3. Waktu Loading-Unloading Exhaust Mazda 323	Lamp B
5.4. Waktu Loading-Unloading Exhaust Mazda MR 90	Lamp B
5.5. Waktu Loading-Unloading Crank Case TS 230	Lamp B
5.6. Waktu Loading-Unloading Crank Case TF 300.....	Lamp B
5.7. Waktu Loading-Unloading Counter Weight F2L-912	Lamp B
5.8. Waktu Loading-Unloading Counter Weight F3L-912	Lamp B
5.9. Waktu Loading-Unloading Counter Weight F4L-912	Lamp B
5.10. Waktu Loading-Unloading Counter Weight F6L-912	Lamp B
5.11. Waktu Loading-Unloading Cylinder Head FL-913	Lamp B
5.12. Waktu Loading-Unloading Cylinder Head FL-912	Lamp B
5.13. Waktu Loading-Unloading Bearing Cap	Lamp B
5.14. Waktu Loading-Unloading Trash Bearing	Lamp B
5.15. Waktu Pergantian Fixture.....	Lamp B
5.16. Hasil Perhitungan Data Intake Mazda 626.....	Lamp B
5.17. Hasil Perhitungan Data Intake Mazda 323	Lamp B
5.18. Hasil Perhitungan Data Exhaust Mazda 323	Lamp B
5.19. Hasil Perhitungan Data Exhaust Mazda MR 90	Lamp B
5.20. Hasil Perhitungan Data Crank Case TS 230	Lamp B
5.21. Hasil Perhitungan Data Crank Case TF 300	Lamp B
5.22. Hasil Perhitungan Data Counter Weight F2L-912	Lamp B
5.23. Hasil Perhitungan Data Counter Weight F3L-912	Lamp B
5.24. Hasil Perhitungan Data Counter Weight F4L-912	Lamp B
5.25. Hasil Perhitungan Data Counter Weight F6L-912	Lamp B
5.26. Hasil Perhitungan Data Cylinder Head FL-913	Lamp B
5.27. Hasil Perhitungan Data Cylinder Head FL-912.....	Lamp B
5.28. Hasil Perhitungan Data Bearing Cap	Lamp B
5.29. Hasil Perhitungan Data Trash Bearing	Lamp B
5.30. Hasil Perhitungan Data Pergantian Fixture.....	Lamp B
6.1. Hasil Penjadwalan Untuk Berbagai Kasus	85
6.2. Input Penjadwalan Model Dasar	87
6.3. Hasil Penjadwalan Model Dasar	88
6.4. Perbandingan Shifting Bottleneck Dengan Priority Dispatching ..	89

6.5.	Input Penjadwalan Kasus Job Tak Siap Pada $t = 0$	92
6.6.	Hasil Penjadwalan Job Tak Siap Pada $t = 0$	92
6.7.	Input Penjadwalan Mesin Tak Siap Pada $t = 0$	96
6.8.	Hasil Penjadwalan Mesin Tak Siap Pada $t = 0$	96
6.9.	Input Penjadwalan Routing Kompleks	99
6.10.	Hasil Penjadwalan Model Routing Kompleks	99
6.11.	Hasil Penjadwalan Di PT. BBI Surabaya	104

BAB I

PENDAHULUAN

1.1. Latar Belakang

Penjadwalan produksi merupakan salah satu aktivitas dari perencanaan produksi yang berhubungan dengan proses pengurutan pengerjaan produk secara menyeluruh pada beberapa mesin atau pengalokasian sumber daya produksi untuk mengerjakan sekumpulan job dalam jangka waktu tertentu.

Sementara itu perencanaan penggunaan sumber daya produksi yang baik yang disertai dengan penjadwalan produksi yang tepat akan menentukan kelancaran produksi pada sistem yang ada. Oleh karena itu perencanaan penjadwalan produksi sangat berpengaruh dalam pelaksanaan produksi sehingga semua perkiraan kebutuhan produksi dapat diperkirakan dengan baik.

Sistem produksi multi produk dengan urutan dan waktu proses yang berbeda tetapi melewati fasilitas mesin-mesin yang sama akan menemui banyak hambatan bila tidak disertai dengan sebuah metoda penjadwalan yang tepat. Karena mungkin saja terjadi penumpukan atau antrian komponen yang berlebihan pada mesin dan waktu menganggur yang berlebihan pada mesin yang lain. Hal ini merupakan salah satu faktor yang besar pengaruhnya terhadap keterlambatan produksi dari perencanaan semula.

Alternatif pemecahan masalah tersebut yang dapat dilakukan tanpa menambah fasilitas produksi adalah dengan menerapkan metoda pengurutan

atau penjadwalan yang tepat untuk sistem yang ada, sehingga produksi bisa selesai tepat pada waktunya bahkan bila memungkinkan sebelum batas waktunya selesai, sehingga waktu yang masih tersisa dapat digunakan untuk menyelesaikan rencana produksi periode berikutnya.

Terdapat banyak metoda atau pendekatan yang dapat digunakan untuk membuat rencana penjadwalan ini. Metoda-metoda atau pendekatan-pendekatan tersebut berbeda satu sama lain, baik masalah yang dipecahkan, cara yang digunakan dan solusi yang dihasilkan serta waktu yang diperlukan untuk memperoleh solusi. Penjadwalan dengan pendekatan optimal adalah perencanaan penjadwalan dengan suatu pendekatan yang memberikan solusi yang terbaik terhadap suatu masalah penjadwalan ditinjau dari kriteria tertentu, tetapi waktu dan kesulitan penyelesaian pendekatan optimal pada masalah yang relatif besar meningkat sesuai dengan tingkat kesulitan masalah. Oleh karena itu dikembangkan pendekatan heuristik yang biasanya memberikan solusi yang lebih cepat dari pendekatan optimal dengan solusi yang cukup memuaskan tetapi tidak menjamin hasil yang optimal. Penelitian terhadap masalah penjadwalan dengan pendekatan heuristik maupun dengan pendekatan optimal terus dilakukan untuk memperoleh solusi yang lebih baik dan atau memberikan waktu penyelesaian masalah yang lebih cepat.

Hampir semua prosedur heuristik penjadwalan job shop yang banyak dijelaskan dalam berbagai literatur merupakan penjadwalan dengan aturan "Priority Dispatching". Prosedur ini berupa aturan-aturan yang digunakan untuk memilih suatu operasi dari sekumpulan operasi tertentu untuk dijadwalkan.

Aturan-aturan tersebut meliputi kriteria seperti SPT (Shortest Processing Time), MWKR (Most Work Remaining), FCFS (First Come First Service) dan lain sebagainya. Kumpulan operasi yang dapat dipilih, dibentuk untuk menghasilkan suatu jadwal aktif, yaitu suatu jadwal dimana tidak ada operasi yang dapat dimulai lebih awal tanpa menunda beberapa operasi yang lain. Dari jadwal aktif ini kemudian akan dihasilkan jadwal non delay, yaitu suatu jadwal dimana sebuah mesin tidak diijinkan dalam keadaan idle pada saat mesin tersebut dapat memulai untuk memproses beberapa operasi. Prosedur heuristik jenis ini menghasilkan solusi yang cepat dan biasanya memberikan solusi yang cukup baik. Dalam banyak situasi pendekatan ini sering dan banyak dibutuhkan dan cukup memadai untuk menyelesaikan masalah penjadwalan, tetapi dengan semakin meningkatnya kecepatan proses perhitungan oleh komputer dan semakin berkembangnya kebutuhan akan penjadwalan yang semakin efisien, maka semakin penting untuk menghasilkan jadwal yang lebih baik dan memberikan solusi yang mendekati optimal atau bahkan optimal dengan mengeluarkan tambahan biaya untuk proses komputasinya.

1.2. Perumusan Masalah

Berdasarkan pertimbangan diatas, maka dalam penelitian ini akan dicoba untuk meneliti dan membuat serta menerapkan metoda pendekatan yang berbeda dengan teknik priority disptching untuk menyelesaikan masalah penjadwalan job shop dengan kriteria minimasi makespan.

Metoda yang akan dikaji dan dicoba untuk diterapkan dalam hal ini adalah metoda Shifthing Bottleneck¹ untuk penjadwalan n job m mesin pada sistem produksi job shop. Dalam pemakaiannya metoda ini didasarkan pada penjadwalan satu mesin². Kemudian juga disertakan teori graph dalam penggambaran masalah penjadwalan job shop, algoritma perhitungan lintasan terpanjang yang didapat dengan memodifikasi algoritma untuk pencarian ruote terpendek, algoritma Schrage dan metoda cabang dan batas untuk penjadwalan satu mesin yang optimal.

Kemudian juga dilakukan pengembangan terhadap model dasar untuk menangani masalah penjadwalan job shop dengan kasus-kasus seperti :

1. Waktu siap atau ready time dari job yang tidak sama dengan nol.
2. Adanya mesin yang tidak siap pada awal periode pelaksanaan penjadwalan.
3. Adanya routing dari job yang kompleks.
4. Adanya job-job yang berprioritas.

1.3. Tujuan dan Manfaat Penelitian

Dalam penelitian kali ini penulis mempunyai tujuan sebagai berikut :

1. Mengetahui performansi metoda Shifting Bottleneck sehingga dapat dipakai sebagai alternatif metoda pembuatan jadwal produksi untuk sistem job shop dengan kriteria minimasi makespan.

¹ Joseph Adams, Egon Balas and Daniel Zawack, *The Shifting Bottleneck Procedure For Job Shop Scheduling*, Management Science Vol. 34, No. 3, March 1988, 391-401

² Jacques Carlier, *The one machine sequencing problem*, Eurpean Journal of Operational Research 11 (1982) 42-47

2. Mengembangkan metoda dasar Shifting Bottleneck untuk menangani permasalahan yang lebih luas dan kompleks.
3. Membuat program komputer untuk problem penjadwalan job shop n job m mesin dengan menggunakan metoda Shifting Bottleneck.

Sedangkan manfaat yang diharapkan dari penelitian ini adalah sebagai berikut :

1. Mengimplementasikan ilmu-ilmu yang diperoleh penulis selama perkuliahan pada permasalahan industri yang nyata, khususnya permasalahan penjadwalan job shop.
2. Mengembangkan dan mengaplikasikan metoda Shifting Bottleneck dalam bidang penjadwaln produksi.
3. Memberikan suatu usulan metoda penjadwalan yang baru bagi perusahaan sebagai suatu alternatif lain dalam membuat jadwal produksi.

1.4. Batasan Masalah

Guna membatasi pembahasan masalah yang diteliti agar tidak terlalu meluas, maka penulis memberikan batasan-batasan atau asumsi-asumsi sebagai berikut :

1. Sistem produksi adalah job shop.
2. Urutan serta waktu proses dari setiap job sudah ditentukan.
3. Setiap mesin hanya dapat memproses satu job pada suatu saat.
4. Waktu siap job dan mesin tidak harus sama dengan nol dan waktu siap ini diketahui secara deterministik.

BAB V PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini berisi data-data mengenai macam dan jumlah job yang dikerjakan, mesin yang dipakai, urutan proses job dan waktu proses job. Kemudian data mentah yang diperoleh tersebut diolah sehingga didapatkan data yang siap digunakan.

BAB VI PENGUJIAN DAN ANALISIS MODEL

Pada bab ini berisi pengujian dan analisis model untuk mengetahui keunggulan dan kelemahan model dan analisis hasil penjadwalan yang diterapkan di PT BBI .

BAB VII KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dan saran yang ditujukan untuk pihak perusahaan dan penulisan karya tulis lebih lanjut.

BAB II

LANDASAN TEORI

Pada bab ini akan dijelaskan serangkaian landasan teori yang mendukung penelitian. Diawali dengan penjelasan mengenai studi pengukuran dan penetapan waktu kerja, pengukuran waktu kerja dengan jam henti, penetapan siklus kerja, test keseragaman data, uji distribusi normal dan serangkaian penjelasan teori dasar mengenai penjadwalan mesin, diantaranya definisi penjadwalan produksi, terminologi dalam penjadwalan dan diakhiri dengan penjelasan tentang teknik priority dispatching rule dalam penjadwalan job shop.

2.1. Studi Pengukuran dan Penetapan Waktu Kerja.

Pada dasarnya pengertian pengukuran kerja adalah metoda penetapan keseimbangan antara jalur manusia yang dikontribusikan dengan unit output yang dihasilkan. Pengukuran kerja ini diperlukan untuk menetapkan waktu baku penyelesaian suatu pekerjaan. Waktu baku merupakan waktu yang dibutuhkan oleh seorang pekerja yang memiliki tingkat kemampuan rata-rata untuk menyelesaikan suatu pekerjaan. Di dalamnya sudah meliputi kelonggaran waktu yang diberikan dengan memperhatikan situasi dan kondisi pekerjaan yang harus diselesaikan tersebut. Sehingga waktu baku yang dihasilkan dalam aktivitas pengukuran kerja dapat digunakan sebagai alat untuk membuat rencana penjadwalan kerja yang menyatakan berapa lama suatu kegiatan itu harus

berlangsung dan berapa output yang akan dihasilkan serta berapa pula jumlah tenaga kerja yang diperlukan untuk menyelesaikan pekerjaan tersebut.

Pada garis besarnya tehnik-tehnik pengukuran waktu kerja ini dapat dibagi kedalam dua bagian, yaitu pengukuran waktu kerja secara langsung dan pengukuran waktu kerja secara tidak langsung. Cara pertama dilakukan dengan melakukan pengamatan secara langsung yaitu di tempat dimana pekerjaan yang diukur dijalankan. Dua cara yang termasuk didalamnya adalah cara pengukuran kerja dengan menggunakan jam henti (stopwatch time-study) dan sampling kerja (work sampling). Cara kedua dilakukan dimana pengamat tanpa berada ditempat pekerjaan yang diukur. Cara ini bisa dilakukan dengan dalam aktivitas waktu baku (standard data) dan data waktu gerakan (predetermined time system).

2.1.1. Pengukuran Waktu Kerja Dengan Jam Henti.

Metoda ini terutama sekali baik diterapkan untuk pekerjaan yang berlangsung singkat dan berulang-ulang (repetitive). Dari hasil pengukuran, maka akan diperoleh waktu baku untuk menyelesaikan satu siklus pekerjaan. Secara garis besar langkah-langkah yang perlu ditempuh dalam metoda ini adalah sebagai berikut :

- Pilih dan definisikan pekerjaan yang akan diukur dan akan ditetapkan waktu standardnya.
- Informasikan maksud dan tujuan pengukuran kerja kepada supervisor atau pekerja.

- Pilih operator dan catat semua data yang berkaitan dengan sistem operasi kerja.
- Bagi siklus kegiatan yang berlangsung ke dalam elemen-elemen kegiatan sesuai dengan aturan yang ada.
- Laksanakan pengamatan dan pengukuran waktu sejumlah N pengamatan untuk setiap siklus atau elemen kegiatan.
- Tetapkan performance rating dari kegiatan yang ditunjukkan operator.
- Laksanakan test kecukupan data dan keseragaman data.
- Tetapkan waktu longgar (allowance time) guna memberikan fleksibilitas.
- Tetapkan waktu kerja baku (standard time) yaitu jumlah total antara waktu normal dan waktu longgar.

Disini juga akan berlaku asumsi-asumsi dasar sebagai berikut :

- ⇒ Metoda dan fasilitas untuk menyelesaikan pekerjaan harus sama dan dibakukan terlebih dahulu sebelum kita mengaplikasikan waktu baku ini untuk pekerjaan yang serupa.
- ⇒ Operator harus memahami benar prosedur dan metoda pelaksanaan kerja sebelum dilakukan pengukuran kerja. Operator-operator yang akan dibebani dengan waktu baku ini, diasumsikan memiliki ketrampilan dan kemampuan yang sama dan sesuai untuk pekerjaan tersebut.
- ⇒ Kondisi lingkungan fisik pekerjaan juga relatif tidak jauh berbeda dengan kondisi fisik pada saat pengukuran kerja dilakukan.
- ⇒ Performance kerja mampu dikendalikan pada tingkat yang sesuai untuk seluruh periode kerja yang ada.

Aktivitas pengukuran kerja dengan jam henti umumnya diaplikasikan pada industri manufakturing yang memiliki karakteristik kerja yang berulang-ulang, terspesifikasi jelas dan menghasilkan output yang relatif sama. Meskipun demikian aktivitas ini dapat juga dilakukan untuk pekerjaan lain dengan kriteria-kriteria sebagai berikut :

- ◆ Pekerjaan tersebut harus dilakukan secara repetitive atau berulang-ulang.
- ◆ Isi atau macam pekerjaan itu harus homogen.
- ◆ Hasil kerja atau output harus dapat dihitung secara nyata (kuantitatif) baik secara keseluruhan ataupun untuk tiap-tiap elemen kerja yang berlangsung.
- ◆ Pekerjaan tersebut cukup banyak dilaksanakan dan teratur sifatnya sehingga akan memadai untuk diukur dan dihitung waktu bakunya.

Ada tiga metoda yang umum digunakan untuk mengukur elemen-elemen kerja dengan menggunakan jam henti, yaitu pengukuran waktu secara terus menerus (continuous timing), secara berulang-ulang (repetitive timing) dan secara penjumlahan (accumulative timing).

Pada pengukuran waktu secara terus menerus, maka pengamat kerja akan menekan tombol stop-watch pada saat elemen kerja pertama dimulai dan membiarkan jarum penunjuk stop-watch berjalan secara terus menerus sampai periode atau siklus kerja selesai berlangsung.

Untuk pengukuran waktu secara berulang-ulang, jarum penunjuk stop-watch akan selalu dikembalikan lagi ke posisi nol pada setiap akhir dari elemen kerja yang diukur. Setelah dilihat dan dicatat waktu kerja diukur kemudian

tombol ditekan lagi dan segera jarum penunjuk bergerak untuk mengukur elemen kerja berikutnya.

Sementara itu metoda pengukuran waktu secara akumulatif memungkinkan pembacaan data waktu secara langsung untuk masing-masing elemen kerja yang ada. Disini akan digunakan dua atau lebih stop-watch yang akan bekerja secara bergantian. Dua atau tiga stop-watch dalam hal ini akan dilekatkan sekaligus pada papan pengamatan dan dihubungkan dengan suatu tuas.

2.1.2. Penetapan Jumlah Siklus Kerja Yang Harus Diukur

Pada dasarnya aktivitas pengukuran kerja merupakan proses sampling. Akibatnya adalah bahwa semakin besar jumlah siklus kerja yang diamati/diukur maka akan semakin mendekati kebenaran akan data waktu yang diperoleh. Konsistensi dari hasil pengukuran dan pembacaan waktu oleh stop-watch akan merupakan hal yang diinginkan dalam proses pengukuran kerja. Semakin kecil variasi data waktu yang ada, jumlah pengamatan atau pengukuran yang harus dilakukan juga akan cukup kecil. Sebaliknya semakin besar variabilitas dari data waktu pengukuran akan menyebabkan jumlah siklus kerja yang diamati juga akan semakin besar agar bisa diperoleh ketelitian yang dikehendaki.

Berikut ini akan diberikan rumus untuk mengevaluasi kesalahan atau penyimpangan terhadap nilai waktu rata-rata dari suatu elemen kerja untuk sejumlah siklus pengukuran atau pengamatan. Diasumsikan bahwa variasi nilai

waktu satu siklus pengamatan ke siklus pengamatan yang lainnya adalah disebabkan oleh faktor-faktor yang serba kebetulan (chance factor). Standard error dari harga rata-rata untuk setiap elemen kerja dapat dinyatakan dalam rumus :

$$\delta_{\bar{x}} = \frac{\delta'}{\sqrt{N}}$$

dimana :

$\delta_{\bar{x}}$ = penyimpangan standard dari distribusi rata - rata.

δ' = penyimpangan standard dari populasi untuk elemen kerja yang ada.

N = jumlah pengamatan untuk elemen kerja yang diukur.

Penyimpangan standard yang dinyatakan dengan tanda (sigma) dapat dirumuskan sebagai berikut :

$$\delta = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}{N}} = \sqrt{\frac{\sum (x - \bar{x})^2}{N}} = \sqrt{\frac{\sum x^2}{N} - \bar{x}^2}$$

$$\delta = \sqrt{\frac{\sum x^2}{N} - \left(\frac{\sum x}{N}\right)^2} = \frac{1}{N} \sqrt{N \sum x^2 - (\sum x)^2}$$

Selanjutnya dengan menggabungkan rumus-rumus yang ada, kita peroleh

$$\delta_{\bar{x}} = \frac{\frac{1}{N} \sqrt{N \sum x^2 - (\sum x)^2}}{\sqrt{N}}$$

Kemudian guna mendapatkan berapa jumlah pengamatan yang seharusnya dibuat (N'), maka disini harus ditetapkan terlebih dahulu berapa tingkat kepercayaan (confidence level) dan derajat ketelitian (degree of accuracy) untuk pengukuran kerja ini. Di dalam aktivitas pengukuran kerja biasanya akan

diambil 95% confidence level dan 5% degree of accuracy. Dengan demikian rumus diatas dapat dinyatakan kembali sebagai berikut :

$$0,05\bar{x} = 2\delta_x$$

$$0,05\frac{\sum x}{N} = 2\frac{\frac{1}{N}\sqrt{N\sum x^2 - (\sum x)^2}}{\sqrt{N'}}$$

$$N' = \left(\frac{40\sqrt{N\sum x^2 - (\sum x)^2}}{N} \right)^2$$

N' adalah jumlah pengamatan atau pengukuran yang seharusnya dilaksanakan untuk memberikan tingkat kepercayaan 95% dan derajat ketelitian 5% dari data waktu yang diukur.

2.1.3. Test Keseragaman Data

Selain kecukupan data harus dipenuhi dalam pelaksanaan time study, maka yang tidak kalah pentingnya adalah bahwa data yang diperoleh haruslah juga seragam. Test keseragaman data perlu kita lakukan terlebih dahulu sebelum kita menggunakan data yang diperoleh guna menetapkan waktu standard. Test keseragaman data bisa dilakukan dengan cara visual dan/atau menerapkan peta kontrol.

Test keseragaman data secara visual dapat dilakukan secara sederhana, mudah dan cepat. Disini kita hanya sekedar melihat data yang terkumpul dan selanjutnya mengidentifikasi data yang terlalu ekstrim. Data ekstrim adalah data yang terlalu besar atau terlalu kecil dan menyimpang jauh dari ternd rata-

ratanya. Kemudian data ekstrim ini kita buang dan tidak dimasukkan dalam perhitungan selanjutnya.

Peta kontrol (control chart) adalah suatu alat yang tepat guna mengetest keseragaman data dan/atau keajegan data yang diperoleh dari pengamatan. Batas kontrol atas (BKA) serta batas kontrol bawah (BKB) dapat dicari dengan rumus sebagai berikut :

$$BKA = \bar{\bar{x}} + 3SD \text{ dan } BKB = \bar{\bar{x}} - 3SD$$

dimana $\bar{\bar{x}} = \bar{\bar{x}}$ dari group.

2.1.4. Uji Kesesuaian Distribusi Normal

Guna mengetahui apakah data-data yang diperoleh dari pengamatan berdistribusi normal atau tidak, maka kita perlu mengujinya dengan uji kebaikan suai. Uji kebaikan suai distribusi normal ini dapat dilakukan dengan dua cara, yaitu Uji Chi Square dan Uji Kolmogorov-Smirnov.

Uji Chi Square dipakai untuk menguji kebaikan suai antara frekuensi harapan dengan frekuensi amatannya yang didasarkan pada besaran :

$$\chi^2 = \sum_{i=1}^k \frac{(o_i - e_i)^2}{e_i}$$

dimana :

χ^2 = nilai peubah acak X^2 yang distribusi sampelnya dihipotesis amat dekat dengan distribusi khi - kuadrat.

o_i = frekuensi amatan dalam sel ke - i.

e_i = frekuensi harapan dalam sel ke - i

Bila frekuensi amatan dekat dengan frekuensi harapannya, maka nilai χ^2 akan kecil, menunjukkan kesesuaian yang baik. Bila frekuensi amatan cukup

berbeda dengan frekuensi harapan maka nilai χ^2 akan besar dan kesesuaian menjadi jelek. Kesesuaian yang baik akan mendukung penerimaan H_0 , sedangkan kesesuaian yang jelek mendukung penolakannya. Daerah kritis akan terjadi pada ujung kanan distribusi khi kuadrat. Untuk taraf kebarartian α , carilah nilai kritis χ_{α}^2 , maka $X^2 > \chi_{\alpha}^2$ menyatakan daerah kritis. Besarnya derajat kebebasan disini tergantung pada dua faktor, yaitu : banyaknya sel dalam percobaan dan banyaknya besaran yang diperoleh dari data amatan yang diperlukan dalam perhitungan frekuensi harapan.

Uji Kolmogorov-Smirnov digunakan untuk data kontinyu dan berdistribusi kontinyu pula. Uji K-S mempunyai keuntungan lebih dibandingkan Uji Chi Square dalam hal tidak adanya informasi data yang hilang karena data tidak perlu dikelompokkan pada saat pengujian.

Jika x_1, x_2, \dots, x_n adalah order statistik dari variabel random yang independent dengan distribusi hipotesa $F^n(x)$ dan distribusi empirisnya didefinisikan sebagai $F_n(x)$. Selanjutnya fungsi distribusi empirisnya tersebut dibandingkan dengan suatu fungsi distribusi penduga $\hat{F}^n(x)$ dengan uji hipotesa sebagai berikut :

$H_0 : F^n(x)$ hipotesa dugaan.

$H_1 : \overline{H_0}$

Fungsi distribusi empiris $F_n(x)$ dari data x_1, x_2, \dots, x_n didefinisikan sebagai $F_n(x_i)$, dimana $i = 1, 2, \dots, n$. Statistik untuk uji K-S adalah sebagai berikut :

$$Dn^+ = \max_{1 \leq i \leq n} \left(\frac{1}{N} - \hat{F}(x_i) \right)$$

$$Dn^- = \max_{1 \leq i \leq n} \left(\hat{F}(x_i) - \frac{i-1}{N} \right)$$

$$Dn = \max(Dn^+, Dn^-)$$

$$H_0 \text{ akan ditolak bila } \left[\sqrt{N} - 0,01 + \frac{0,85}{\sqrt{N}} \right] Dn > C_{1-\alpha}$$

dimana $C_{1-\alpha}$ didapat dari tabel Kolmogorov-Smirnov. Data berdistribusi normal jika H_0 diterima. \neq

2.1.5. Penetapan Waktu Baku

Waktu normal untuk suatu elemen operasi kerja adalah semata-mata menunjukkan bahwa seorang operator yang berkualifikasi baik akan bekerja menyelesaikan pekerjaan pada kecepatan yang normal. Walaupun demikian pada kenyataannya kita akan melihat bahwa tidaklah bisa diharapkan operator tersebut mampu bekerja secara terus-menerus sepanjang hari tanpa adanya interupsi sama sekali. Kenyataannya operator akan sering menghentikan kerja dan membutuhkan waktu-waktu khusus untuk keperluan seperti personal needs, istirahat melepas lelah dan alasan-alasan lain yang diluar kontrolnya. Waktu baku yang akan ditetapkan harus mencakup semua elemen-elemen kerja dan ditambah dengan kelonggaran-kelonggaran (allowance) yang perlu dan biasanya kelonggaran waktu ini merupakan sekian prosen dari waktu normal. Sehingga waktu baku disini adalah waktu normal ditambah dengan waktu longgar yang selanjutnya dapat dinyatakan sebagai berikut :

$$\text{Waktu Baku} = \text{Waktu Normal} + (\text{Waktu Normal} \times \% \text{Allowance})$$

$$\text{Waktu Baku} = \text{Waktu Normal} \times \frac{100\%}{100\% - \text{Allowance}}$$

dan

$$\text{Waktu Normal} = \text{Waktu Observasi Rata-rata} \times \text{Performance Rating.}$$

2.2. Definisi Penjadwalan Produksi

Penjadwalan produksi didefinisikan sebagai pengalokasian sumber daya produksi (mesin dan operator), kapan mulai operasi dan kapan selesai operasi untuk mengerjakan sejumlah job. Hal ini berarti penjadwalan produksi juga sebagai rencana pengurutan kerja serta pengalokasian sumber daya produksi baik waktu maupun fasilitas untuk setiap operasi yang harus diselesaikan. Penjadwalan juga dapat diartikan sebagai proses pengalokasian sumber-sumber guna melaksanakan sekumpulan tugas dalam jangka waktu tertentu¹.

Upp
08 11 01 09 16 / 2011

2.2.1. Penjadwalan Produksi

Permasalahan penjadwalan job dapat dinyatakan sebagai berikut²:

“ Ada sekumpulan n job yang akan diproses, dimana tiap job memiliki waktu setup, waktu proses dan due date. Untuk dapat menyelesaikannya, tiap job diproses pada beberapa mesin. Dalam hal ini diperlukan untuk menjadwalkan

¹ Kenneth R. Baker., Introduction To Sequencing and Scheduling, John Wiley & Sons Inc, New York London Sydney Toronto, 1974, hal 2

² Elsayed, Elsayed A., Analysis and Control of Production System, Prentice Hall, New Jersey, 1985, hal 226

job-job tersebut pada mesin-mesin guna dapat mengoptimalkan kriteria performansi tertentu “.

Dari pengertian permasalahan penjadwalan job diatas, kita dapat menyatakan bahwa tujuan dari penjadwalan job adalah untuk mengoptimalkan kriteria performansi tertentu dan kriteria yang dioptimalkan ini dapat berupa kriteria performansi tunggal atau gabungan dari beberapa kriteria performansi tertentu.

Terdapat beberapa kriteria performansi yang dapat digunakan untuk mengevaluasi ukuran keberhasilan dari suatu penjadwalan. Kriteria itu adalah sebagai berikut :

- Makespan, yaitu total waktu yang diperlukan untuk menyelesaikan seluruh job.

$$M = \sum_{i=1}^n t_i$$

dimana t_i = waktu proses job ke i

M = makespan

- Mean Flow Time, yaitu waktu rata-rata sebuah job berada di dalam sistem.

$$\bar{F} = \frac{\sum_{i=1}^n F_i}{n}$$

$$F_i = C_i - r_i$$

dimana \bar{F} = mean flow time

F_i = flow time job ke i

C_i = completion time job ke i

r_i = ready time job ke i

- Mean Lateness, yaitu rata-rata beda completion time suatu job dengan due datenya atau rata-rata beda waktu antara waktu penyelesaian suatu job dengan batas akhirnya.

$$\bar{L} = \frac{\sum_{i=1}^n L_i}{n}$$

$$L_i = C_i - d_i$$

dimana \bar{L} = mean lateness

L_i = lateness job ke i

d_i = due date job ke i

- Mean Tardiness, yaitu rata-rata keterlambatan penyelesaian (positive lateness) suatu job.

$$\bar{T} = \frac{\sum_{i=1}^n T_i}{n}$$

$$T_i = \max(0, C_i - d_i)$$

dimana T_i = tardiness job ke i

- Number of Tardy Job, yaitu jumlah job yang terlambat.

$$N_T = \sum_{i=1}^n \delta_i$$

$$\delta_i = 1 \text{ jika } T_i > 0$$

$$\delta_i = 0 \text{ jika } T_i \leq 0$$

- Maximum Lateness, yaitu nilai maksimum dari lateness.

$$L_{\max} = \max(L_i)$$

- Weighted Mean Flow Time, yaitu sama dengan mean flow time, tetapi dengan mempertimbangkan prioritas atau bobot masing-masing job.

$$\overline{F}_w = \frac{\sum_{i=1}^n w_i F_i}{\sum_{i=1}^n w_i}$$

dimana w_i = bobot atau prioritas job ke i

- Weighted Mean Tardiness, yaitu sama dengan mean tardiness, tetapi dengan mempertimbangkan bobot atau prioritas masing-masing job.

$$\overline{T}_w = \frac{\sum_{i=1}^n w_i T_i}{\sum_{i=1}^n w_i}$$

- Mean Earliness, yaitu waktu rata-rata sebuah job yang diselesaikan sebelum due datenya.

$$\overline{E} = \frac{\sum_{i=1}^n E_i}{n}$$

$$E_i = \min(0, C_i - d_i)$$

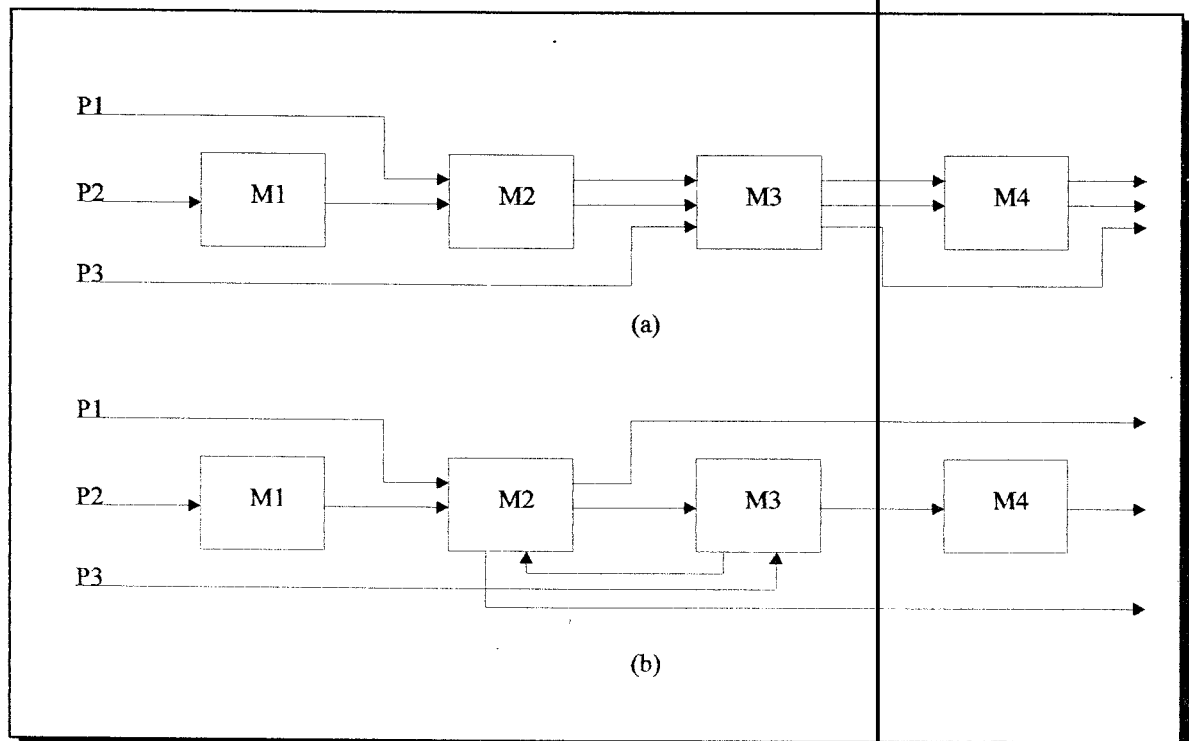
dimana \overline{E} = mean earliness

E_i = earliness atau negative lateness dari job ke i

Sebelum membuat suatu jadwal produksi, kita memerlukan sekumpulan informasi dan informasi ini menjadi faktor yang menentukan jenis atau tipe permasalahan penjadwalan yang akan dibuat. Informasi-informasi itu meliputi :

1. Jumlah job yang akan dijadwalkan. Informasi ini termasuk waktu yang dibutuhkan untuk tiap proses dan jenis mesin yang akan dipakai.

2. Jumlah mesin yang akan digunakan. Informasi ini menentukan apakah penjadwalan yang akan dibuat merupakan penjadwalan mesin tunggal (single machine) atau banyak mesin (multiple machine).
3. Tipe atau jenis fasilitas manufakturing. Informasi ini menentukan apakah penjadwalan yang akan dibuat merupakan penjadwalan untuk sistem produksi flow shop atau job shop.
 - Flow shop berarti sistem produksi yang memiliki aliran proses atau urutan permesinan yang satu arah untuk semua job.
 - Job shop berarti sistem produksi dimana aliran proses atau urutan permesinan untuk semua job tidak sama atau alirannya acak.



Gambar 2.1.
(a). Aliran proses flow shop (b). Aliran proses job shop

4. Pola kedatangan job. Pola kedatangan job ke dalam sistem produksi dua macam, yaitu statis atau dinamis.

- Statis, artinya job-job harus ada untuk diproses pada awal periode penjadwalan dan tidak ada kedatangan job baru selama periode produksi.
- Dinamis, artinya job-job datang berturut-turut selama periode produksi menurut proses stokastik.

Sistem penjadwalan dengan pola kedatangan job dinamis lebih rumit dibandingkan dengan pola kedatangan job statis, oleh karena itu biasanya digunakan simulasi komputer.

5. Kriteria performansi yang diinginkan. Kriteria yang diinginkan menentukan ukuran keberhasilan performansi dari suatu jadwal yang akan dibuat dan kriteria itu antara lain makespan, mean tardiness, mean flow time dan lain-lain.

2.2.2. Terminologi Dalam Penjadwalan

Terminologi atau istilah-istilah yang sering digunakan dalam penjadwalan adalah sebagai berikut :

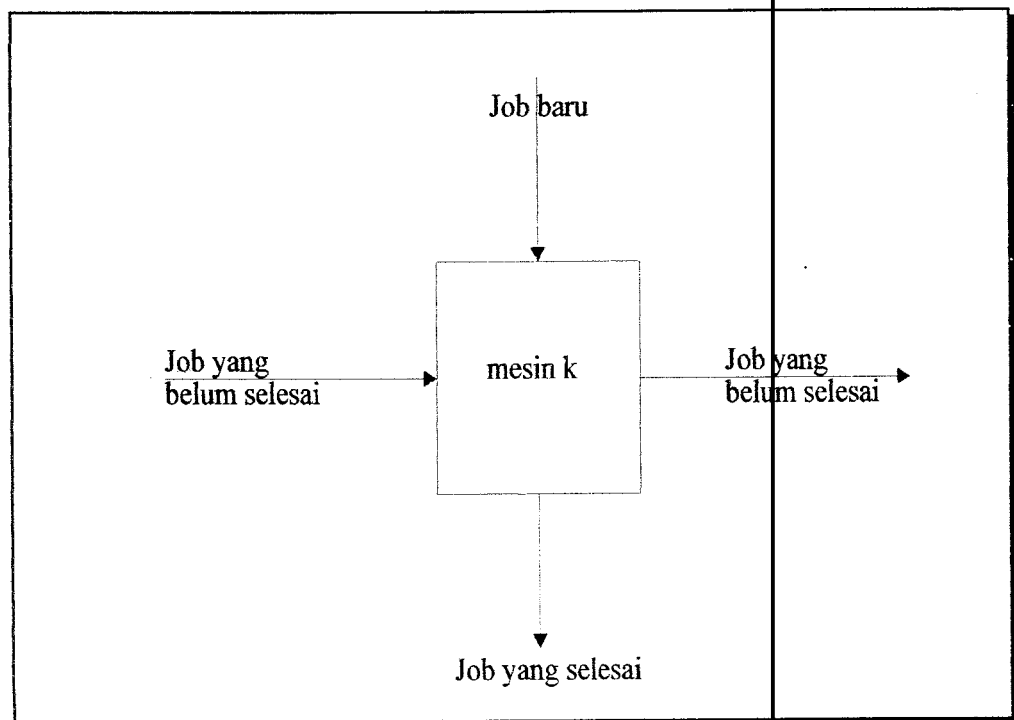
1. Processing time (t_i), yaitu perkiraan lamanya waktu yang dibutuhkan untuk menyelesaikan job i . Didalamnya juga termasuk waktu setup.
2. Ready time (r_i), yaitu saat job i siap dijadwalkan.
3. Due date (d_i), yaitu batas waktu penyelesaian yang ditetapkan untuk suatu job i , dimana lebih dari itu maka dianggap terlambat.
4. Completion time (C_i), yaitu saat job i selesai dikerjakan.

5. Flow time (F_i), yaitu lamanya job i berada di rantai produksi. Flow time dihitung sejak job siap dijadwalkan sampai dengan job selesai dikerjakan.
6. Lateness (L_i), yaitu penyimpangan antara completion time job i dengan due datenya. Lateness job i dapat bernilai positif atau negatif. Jika sebuah job diselesaikan tepat atau setelah due datenya, maka mempunyai positive lateness. Tetapi jika job i itu diselesaikan sebelum due datenya, maka mempunyai negative lateness (earliness).
7. Tardiness (T_i), yaitu nilai dari positive lateness sebuah job i . Jika sebuah job diselesaikan lebih awal dari due datenya, maka job tersebut akan mempunyai negative lateness, tetapi nol tardiness. Jika sebuah job mempunyai positive lateness, maka job itu akan mempunyai harga tardiness sama dengan positive latenessnya.
8. Slack (S_i), yaitu waktu yang tersisa (remaining time) antara due date dengan processing timenya.
9. Heuristic, yaitu prosedur atau aturan untuk menyelesaikan suatu masalah dengan menghasilkan solusi yang baik, tetapi tidak menjamin hasil yang optimal.

2.2.3. Penjadwalan Job Shop

Dalam satu aspek penting, problem klasik penjadwalan job shop akan berbeda dengan problem flow shop, yaitu aliran kerjanya yang tidak satu arah. Elemen yang ada adalah sejumlah mesin dan beberapa job yang akan dijadwal. Masing-masing job terdiri dari beberapa operasi dengan struktur linier

precedence yang sama seperti pada model flow shop. Formulasi yang paling umum untuk problem job shop adalah masing-masing job mempunyai operasi sebanyak m dan tiap operasi dikerjakan pada satu mesin. Tidak seperti model pada flow shop, tidak ada mesin yang paling awal yang digunakan untuk memproses hanya pada operasi pertama dari sebuah job atau tidak ada mesin yang paling akhir yang digunakan untuk memproses hanya pada operasi terakhir dari sebuah job. Hal ini dapat dilihat pada gambar 2.2. yang menerangkan aliran kerja pada suatu mesin dalam model job shop.

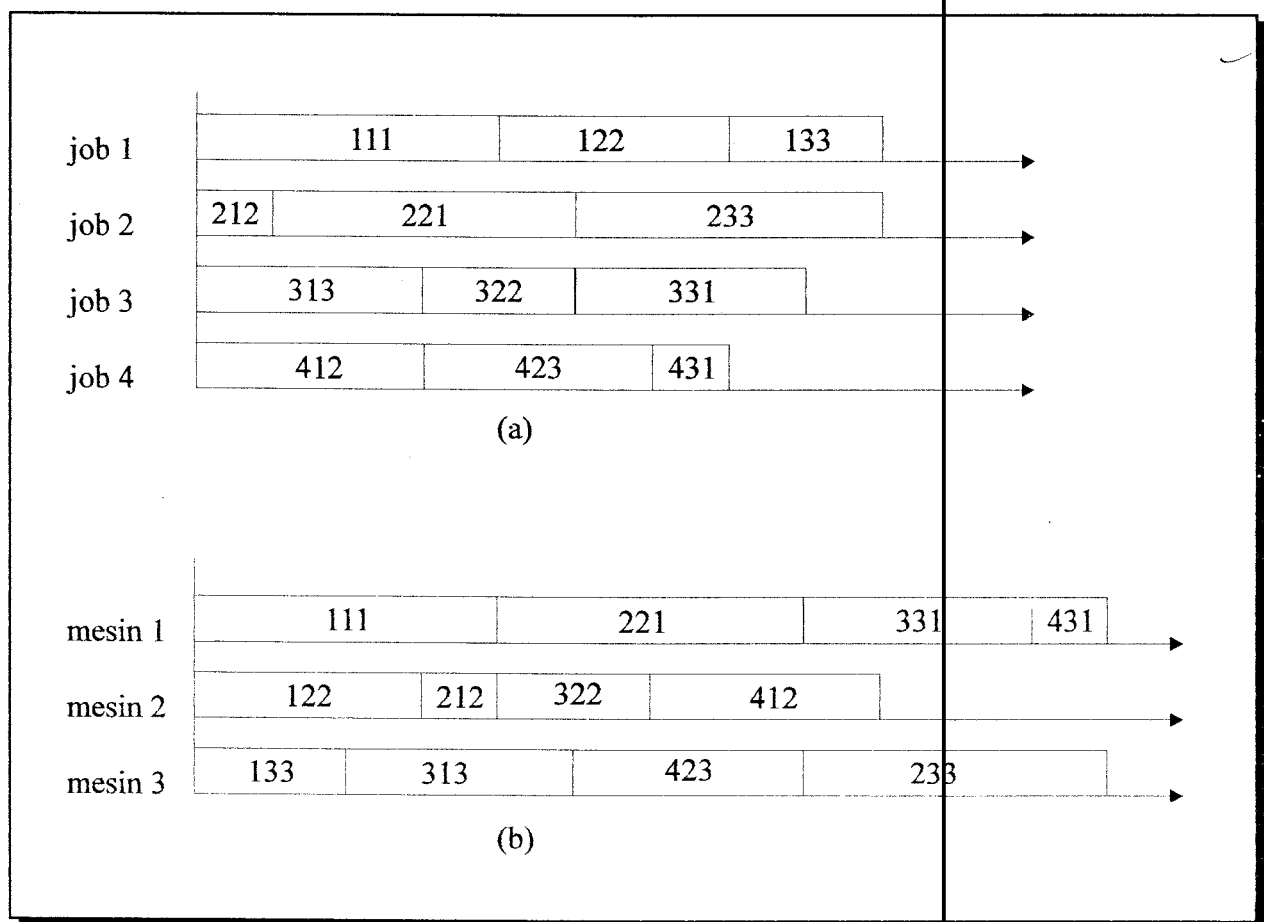


Gambar 2.2. Aliran kerja pada mesin job shop.

Dalam model flow shop operasi k dari suatu job dilakukan oleh mesin k dan tidak diperlukan untuk membedakan antara nomer operasi dan nomer mesin.

Dalam kasus job shop, guna menggambarkan suatu operasi dinyatakan dengan notasi (i,j,k) yang artinya job i dengan operasi ke j memerlukan mesin k.

Problem penjadwalan job shop secara grafik dapat dinyatakan dalam dua bentuk, yaitu job by job atau machine by machine, seperti yang terlihat dalam gambar 2.3.



Gambar 2.3. Penggambaran secara grafik problem job shop.

(a). Job by job. (b). Machine by machine

Penyajian data-data yang diperlukan dalam problem job shop dapat ditabelkan dalam tabel routing, yaitu penugasan operasi mesin dan tabel waktu proses.

		Operasi		
		1	2	3
Job	1	1	2	3
	2	2	1	3
	3	3	2	1
	4	2	3	1

(a)

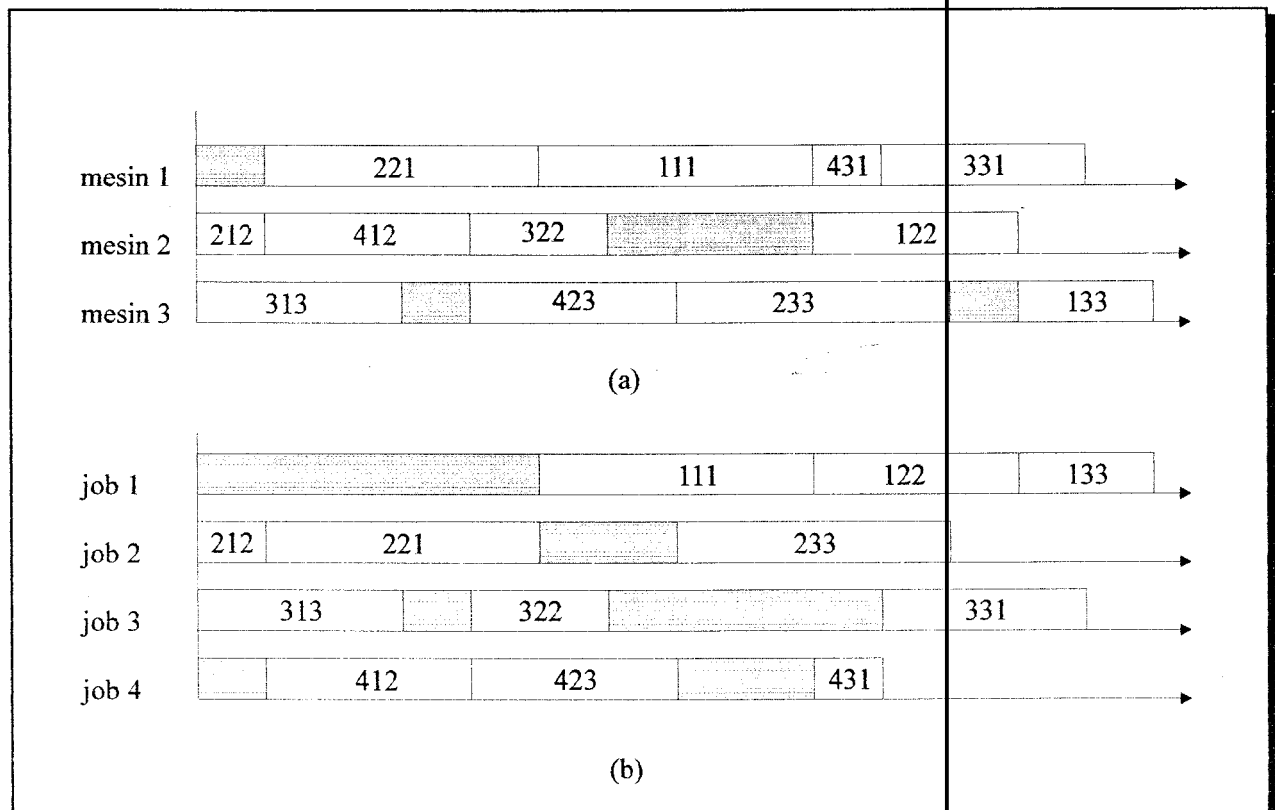
		Operasi		
		1	2	3
Job	1	4	3	2
	2	1	4	4
	3	3	2	3
	4	3	3	1

(b)

Tabel 2.1.

(a). Routing (b). Waktu proses

Salah satu penyelesaian yang layak dari problem penjadwalan job shop diatas dapat digambarkan berdasarkan mesin-mesin (Gant Chart) atau bisa juga berdasarkan job-job yang dikerjakan, seperti terlihat pada gambar 2.4.



Gambar 2.4. Dua bentuk jadwal job shop
(a). Gant Chart. (b). Jadwal berdasarkan job.

2.2.4. Jenis-jenis Jadwal

Pada prinsipnya, terdapat jadwal feasible yang tak terhingga jumlahnya, karena sejumlah waktu idle dapat disisipkan pada mesin dua operasi yang berurutan. Penyesuaian waktu dimulainya suatu operasi sama dengan menggeser blok operasi ke kiri pada Gant Chart pada saat mengatur urutan operasi. Jenis penyesuaian ini disebut local left-shift atau limited left-shift. Kumpulan jadwal dimana local left-shift sudah tidak dapat dibuat dinamakan jadwal semiaktif dan ini sama dengan kumpulan jadwal yang tidak mengandung superfluous idle time. Kumpulan jadwal ini mendominasi dari semua jadwal, artinya bahwa untuk mengoptimalkan ukuran-ukuran performansi cukup dengan memperhatikan jadwal semiaktif saja.

Jumlah jadwal semiaktif yang layak terbatas, tetapi jumlahnya cukup banyak. Jika urutan pada mesin-mesin semuanya independen maka akan terdapat jadwal semiaktif sebanyak $(n!)^m$. Sebagai contoh, banyaknya $n = 2$ dan $m = 2$ dimana job mempunyai routing seperti pada tabel 2.2.

		Operasi	
		1	2
Job	1	1	2
	2	2	1

Tabel 2.2.
Routing untuk problem $n = 2$ dan $m = 2$

Meskipun dalam kasus ini terdapat $(2!)^2 = 4$ jadwal semiaktif, tetapi kemungkinan untuk membuat jadwal semiaktif yang feasible hanya tiga buah. untuk lebih jelasnya dapat dilihat pada gambar 2.5.

111			221
	122	212	
(a)			
	221	111	
212			122
(b)			
111		221	
212		121	
(c)			

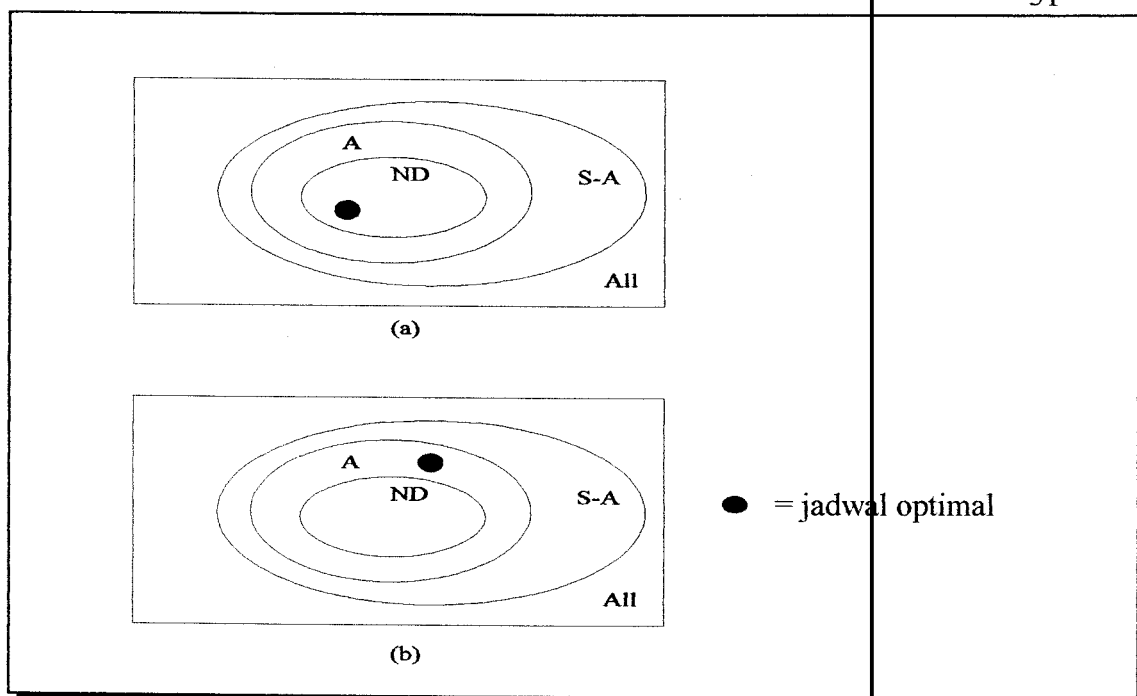
Gambar 2.5.

Penjadwalan semiaktif untuk $n = 2$ dan $m = 2$

Jenis penyesuaian dimana beberapa operasi dapat dimulai lebih awal tanpa menunggu operasi lainnya disebut global left-shift atau hanya left-shift saja. Left-shift mengijinkan sebuah operasi untuk melompati operasi lainnya menuju interval waktu idle jika interval waktu tersebut cukup untuk menampung operasi yang bersangkutan. Sekumpulan jadwal dimana tidak ada global left-

shift yang dapat dibuat dinamakan kumpulan jadwal aktif dan jadwal ini mendominasi jadwal semiaktif, sehingga cukup memperhatikan jadwal aktif saja.

Jadwal non delay merupakan subset dari jadwal aktif dimana tidak mengijinkan adanya mesin yang dibiarkan idle pada saat mesin tersebut dapat memulai untuk memproses beberapa operasi. Semua jadwal non delay merupakan jadwal aktif. Sehingga jumlah jadwal non delay lebih sedikit dibanding jumlah jadwal aktif akan tetapi jadwal non delay tidak mendominasi jadwal aktif. Jadwal non delay tidak memberikan jaminan terdapatnya sebuah jadwal yang optimal. Pada diagram Venn gambar 2.6a terlihat bahwa paling sedikit terdapat satu jadwal optimal adalah jadwal non delay, sedangkan pada diagram venn gambar 2.6b jadwal optimal bukan termasuk jadwal non delay. Meskipun demikian jadwal non delay dapat diharapkan untuk memberikan solusi yang sangat baik.



Gambar 2.6

Diagram Venn hubungan antara jadwal non delay, aktif dan semiaktif

(a) Jadwal optimal merupakan bagian jadwal nondelay

(b) Jadwal optimal diluar jadwal nondelay

2.2.5. Teknik Priority Dispatching

Prosedur Heuristik yang didesain untuk melakukan penjadwalan penuh membutuhkan penyelesain terhadap konflik yang timbul dalam satu cara. Hal ini berarti prosedur tersebut harus mempunyai aturan pengurutan atau priority dispatching dalam menentukan atau memilih satu operasi atau job yang mengalami konflik yang akan diproses selanjutnya. Terdapat beberapa aturan priority dispatching yang dapat dipakai untuk pengurutan job, yaitu :

1. SPT (Shortest Processing Time), yaitu memilih operasi yang mempunyai waktu proses yang paling kecil.

2. EDD (Earliest Due Date), yaitu operasi yang memiliki due date yang paling awal dikerjakan lebih dahulu.
3. FCFS (First Come First Serve), yaitu memilih operasi yang datang lebih awal untuk diproses lebih dahulu.
4. R (Random), yaitu memilih operasi secara acak atau random untuk diproses.
5. MWKR (Most Work Remaining), yaitu operasi yang memiliki sisa waktu proses yang paling banyak mendapat prioritas pertama untuk dikerjakan lebih dahulu.
6. LWKR (Least Work Remaining), yaitu memilih operasi dengan paling sedikit waktu proses yang belum terjadwal.
7. MOPNR (Most Operations Remaining), yaitu memilih operasi yang memiliki operasi pengikut yang terbanyak.

LSKR (Least Slack Remaining), yaitu memilih operasi yang memiliki waktu slack yang terkecil dimana waktu slack dihitung dengan mengurangi waktu proses dari due datenya.

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan langkah-langkah yang ditempuh oleh penulis dalam melakukan penelitian. Langkah-langkah tersebut meliputi penetapan tujuan penelitian, identifikasi dan perumusan masalah, studi literatur, penelitian dan pembuatan model, pengumpulan dan pengolahan data, analisis model kemudian diakhiri dengan kesimpulan dan saran.

3.1. Tujuan Penelitian

Guna mengarahkan penelitian ini, maka ditetapkan terlebih dahulu tujuan dari penelitian itu sendiri. Tujuan penelitian ini menjadi arahan untuk seluruh langkah berikutnya dalam penelitian, sehingga setiap langkah yang dilakukan harus mendukung tercapainya tujuan yang ditetapkan. Tujuan penelitian ini telah dijelaskan dengan rinci pada Bab I di depan.

3.2. Identifikasi dan Perumusan Masalah

Di dalam langkah ini penulis melakukan identifikasi dan perumusan masalah dengan jalan melakukan pengkajian dan penelitian serta kemudian menerapkan metoda pendekatan lain yang berbeda dengan teknik priority dispatching untuk menyelesaikan masalah penjadwalan job shop dengan kriteria minimasi makespan. Metoda pendekatan yang akan dikaji dan dicoba untuk

diterapkan dalam hal ini adalah metoda Shifting Bottleneck yang ide dasarnya didapatkan penulis dari jurnal Management Science Vol. 34, No. 3, March 1988 dengan judul *The Shifting Bottleneck Procedure For Job Shop Scheduling* yang ditulis oleh Joseph Adams, Egon Balas dan Daniel Zawack.

3.3. Studi Literatur

Sebagai landasan dan kerangka berpikir bagi penelitian yang akan dilakukan, maka penulis perlu menyertakan beberapa konsep atau teori yang mendukung penelitian, baik secara langsung maupun tidak langsung. Konsep atau teori yang disertakan antara lain studi pengukuran dan penetapan waktu kerja, pengumpulan dan pengolahan data statistik dan teori dasar mengenai pengurutan dan penjadwalan. Konsep dan teori tersebut didapatkan dari berbagai sumber bacaan yang diperlukan.

3.4. Penelitian dan Pembuatan Model

Di dalam langkah ini dilakukan penelitian dan pembuatan model dasar untuk penyelesaian permasalahan penjadwalan job shop dengan metoda Shifting Bottleneck yang ide dasarnya diambil penulis dari jurnal yang telah disebutkan di muka. Kemudian dilakukan pengembangan model dasar yang telah dibentuk untuk menyelesaikan masalah penjadwalan yang lebih kompleks.

3.5. Pengumpulan dan Pengolahan Data

Sebelum memasuki ke langkah analisis model, penulis memerlukan data-data yang dipakai sebagai input model yang telah dibentuk. Data yang dikumpulkan adalah data sekunder yang diambil dari penelitian mahasiswa sebelumnya. Data-data tersebut meliputi data waktu proses, routing, jumlah mesin dan job-job yang dikerjakan. Kemudian data-data tersebut diolah sebelum dimasukkan ke model.

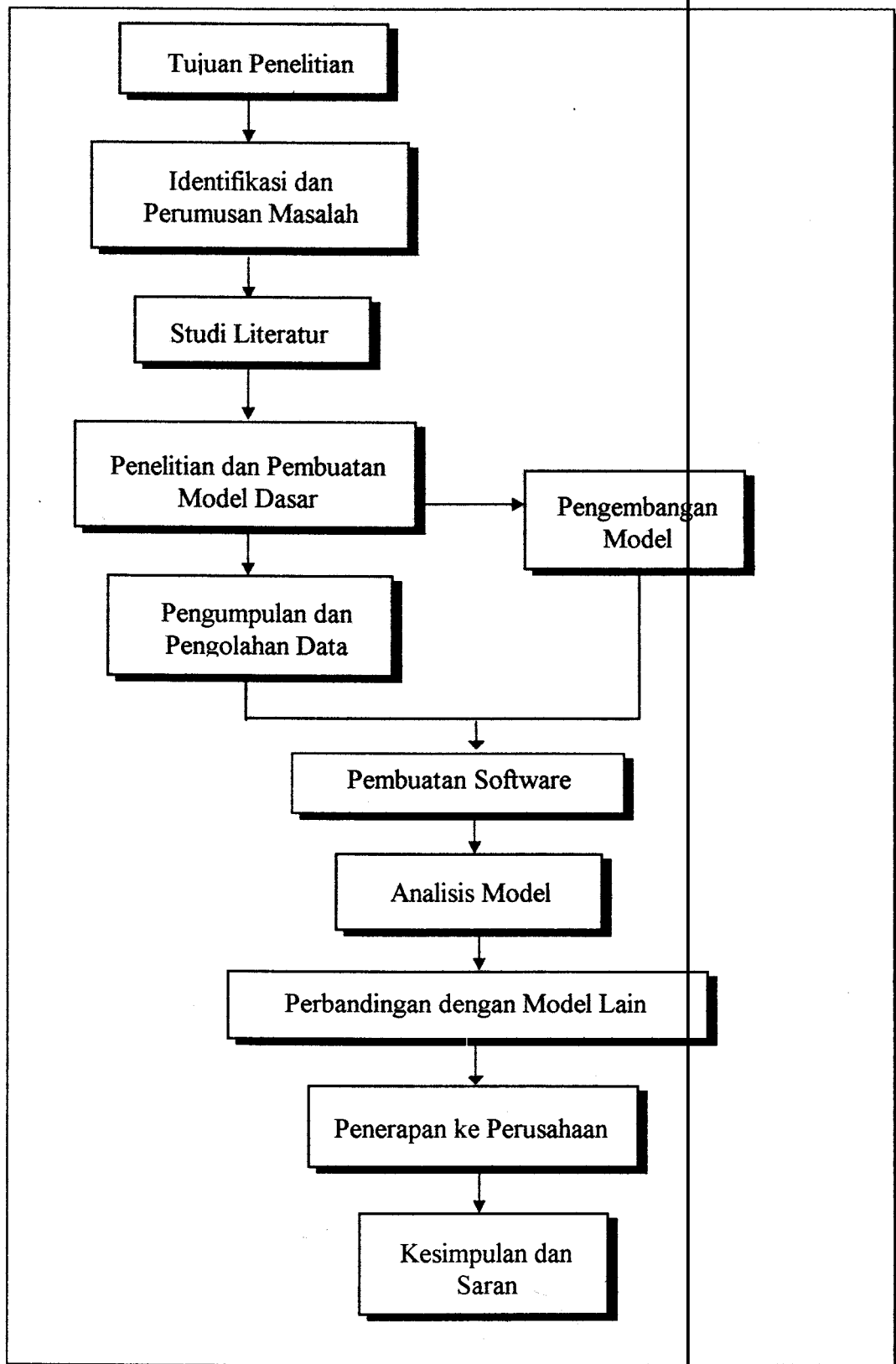
3.6. Analisis Model

Analisis model dilakukan dengan cara membandingkan performansi model yang dikembangkan dengan model lain yang didasarkan teknik priority dispatching yang terdapat dalam paket program Quant System. Performansi yang dibandingkan dalam hal ini adalah waktu proses (running time) yang diperlukan untuk menghasilkan jadwal dan nilai makespan yang dihasilkan masing-masing metoda untuk berbagai macam permasalahan yang berbeda. Selanjutnya juga dicoba untuk diterapkan dengan memakai data dari PT. BBI Surabaya.

3.7. Kesimpulan dan Saran

Dari hasil analisis model kemudian disimpulkan beberapa karakteristik yang ditunjukkan oleh performansi model dan dilanjutkan ke saran yang perlu dilakukan untuk penyempurnaan model.

Secara umum, langkah-langkah yang ditempuh penulis dalam melakukan penelitian dapat dilihat dalam gambar 3.1.



Gambar 3.1. Langkah-langkah Penelitian

BAB IV

PENYUSUNAN DAN PENGEMBANGAN

METODA SHIFTING BOTTLENECK

Pada bab ini akan dibahas mengenai penyusunan algoritma dasar yang terdapat dalam metoda Shifting Bottleneck dan kemudian dicoba untuk mengembangkannya guna menyelesaikan masalah penjadwalan job shop yang lebih kompleks.

Penulis mengambil algoritma dasar yang terdapat dalam metoda Shifting Bottleneck dari penelitian yang telah dilakukan oleh Joseph Adams, Egon Balas dan Daniel Zawack (Management Science, Vol . 34, No. 3, Maret 1988) dan penelitian dari J. Carlier (European Journal of Operation Research 11, 1982, 42-47).

Metoda pendekatan ini digunakan untuk menyelesaikan masalah penjadwalan job shop dengan kriteria minimasi makespan yang dilakukan dengan cara menjadwalkan mesin satu per satu secara berturut-turut. Setiap kali setelah suatu mesin baru dijadwalkan, dilakukan reoptimasi lokal terhadap mesin-mesin yang sebelumnya telah terjadwal. Pencarian suatu mesin yang dianggap bottleneck dan prosedur untuk melakukan reoptimasi lokal, keduanya berdasarkan pada penyelesaian masalah penjadwalan satu mesin secara berulang-ulang.

4.1. Permasalahan

Kita mengetahui, bahwa permasalahan penjadwalan job shop atau pengurutan permesinan pada dasarnya merupakan permasalahan pengaturan urutan job-job yang akan diproses pada mesin-mesin dengan tujuan untuk meminimumkan beberapa fungsi dari completion time dari job-job yang diproses tersebut dengan pembatas, yaitu :

1. Urutan permesinan untuk tiap job telah ditentukan.
2. Tiap mesin hanya dapat memproses satu job pada suatu saat.

Dalam hal ini, masalah yang diselesaikan oleh metoda Shifting Bottleneck adalah masalah penjadwalan job shop, yaitu penjadwalan job-job yang akan diproses pada mesin-mesin dengan tujuan untuk meminimumkan makespan.

Selanjutnya permasalahan tersebut dapat dinyatakan sebagai berikut :

Minimum t_n

$$t_j - t_i \geq d_i \quad (i,j) \in A$$

$$t_i \geq 0 \quad i \in N$$

$$t_j - t_i \geq d_i \vee t_i - t_j \geq d_j \quad (i,j) \in E_k, k \in M \quad (P)$$

dimana :

$N = \{0, 1, \dots, n\}$ merupakan himpunan operasi-operasi yang akan dijadwalkan dengan 0 dan n adalah operasi dummy “start” dan “finish”.


$M =$ himpunan mesin-mesin.


A = himpunan pasangan operasi-operasi yang memenuhi precedence relation.


E_k = himpunan pasangan operasi-operasi yang akan diproses pada mesin k .

Setiap solusi yang layak atau feasible dari (P) merupakan suatu jadwal.

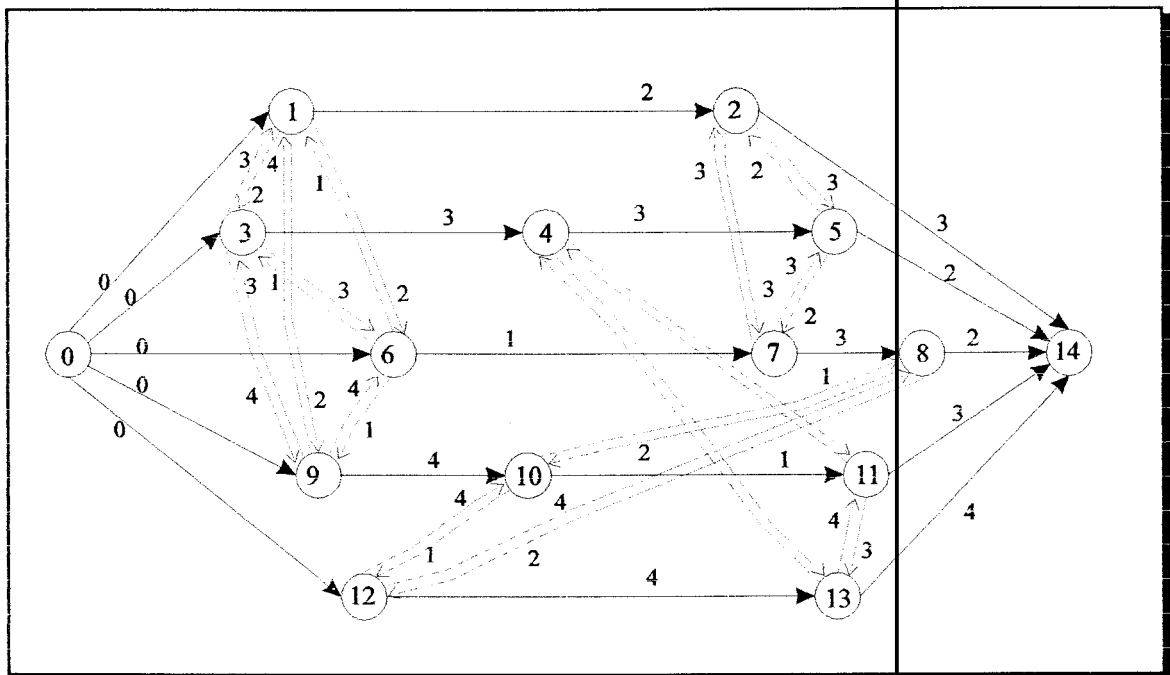
Selanjutnya permasalahan diatas akan lebih jelas jika dinyatakan dalam suatu disjunctive graph $G = (N, A, E)$ dimana

N = himpunan simpul 

A = himpunan busur conjunctive 

E = himpunan busur disjunctive 

Sebagai contoh misalnya disjunctive graph untuk permasalahan job shop dengan 5 job 4 mesin dengan total operasi ada 15 buah, dapat dinyatakan seperti pada gambar 4.1.



Gambar 4.1. Disjunctive Graph 5 Job 4 Mesin.

Dalam disjunctive graph $G = (N, A, E)$, N merupakan himpunan simpul-simpul yang mewakili operasi-operasi dari tiap job, A adalah himpunan busur-busur berarah yang menghubungkan operasi-operasi dari tiap job dan E merupakan himpunan busur-busur disjunctive yang menghubungkan operasi-operasi yang dikerjakan pada mesin yang sama. Angka-angka yang terdapat diatas busur menyatakan waktu proses. Himpunan busur disjunctive E dipecah menjadi kumpulan-kumpulan E_k , $E = \cup (E_k : k \in M)$ dimana tiap mesin hanya satu.

Kemudian dinyatakan $D = (N, A)$ sebagai busur berarah yang diperoleh dari G dengan menghilangkan semua busur disjunctivenya. Pemilihan busur-busur disjunctive dalam E_k akan menghasilkan S_k yang merupakan busur tunggal yang menghubungkan operasi-operasi yang dikerjakan pada mesin k . Busur

tunggal S_k ini akan acyclic jika tidak terdapat siklus berarah dan setiap S_k yang acyclic ini akan membentuk urutan operasi yang unik yang akan dikerjakan oleh mesin k . Jadi pengurutan operasi dalam mesin k merupakan pemilihan busur tunggal yang acyclic dalam E_k . S merupakan himpunan atau pemilihan lengkap dari S_k dimana $k \in M$. Pemilihan sebagian didefinisikan sebagai pemilihan lengkap seperti diatas dari suatu himpunan bagian $M_o \in M$, yang berarti merupakan urutan operasi dari semua mesin elemen M_o . M_o merupakan himpunan mesin-mesin yang telah terjadwal. Pemilihan lengkap dari himpunan busur S , yaitu dengan cara mengganti busur-busur disjunctive anggota E dengan satu busur tunggal anggota himpunan S , akan menghasilkan jaringan D_s , $D_s = (N, A \cup S)$. Selanjutnya jaringan D_s yang terbentuk ini merupakan suatu jadwal yang unik. Kemudian kita mencari suatu pemilihan lengkap yang acyclic $S \subseteq E$ dalam jaringan D_s yang dapat meminimumkan panjang dari lintasan terpanjangnya.

4.2. Pendekatan Penyelesaian

Dalam menyelesaikan permasalahan penjadwalan job shop dengan kriteria minimasi makespan, pendekatan yang diambil dalam metoda Shifting Bottleneck adalah dengan cara menjadwalkan mesin secara optimal satu per satu secara berturut-turut. Guna dapat melakukan penjadwalan ini untuk setiap mesin yang menjadi anggota mesin yang belum terjadwal ($M \setminus M_o$), kita selesaikan penjadwalan satu mesin (one machine scheduling problem). Kemudian solusi yang dihasilkan kita gunakan untuk meranking mesin-mesin

tersebut dan mesin yang menempati ranking tertinggi masuk menjadi anggota mesin yang telah terjadwal (M_o). Setiap kali ada mesin baru yang masuk menjadi anggota mesin yang telah terjadwal, kita lakukan reoptimasi lokal terhadap mesin-mesin yang telah terjadwal sebelumnya.

Kontribusi utama yang diberikan dalam pendekatan ini adalah adanya relaksasi terhadap problem utama, yaitu pada saat memilih mesin mana yang harus dijadwalkan. Pemilihan itu sendiri berdasarkan pada ide klasik dalam memberikan prioritas terhadap mesin-mesin yang dianggap sebagai mesin kritis yang bottleneck.

Terdapat banyak cara dalam menentukan sebuah mesin yang dianggap bottleneck, diantaranya adalah sebagai berikut :

1. **Criticality**, yaitu suatu pemilihan sebagian $S = \cup (S_k : k \in M_o)$ yang berhubungan dengan diagraph D_s . Kemudian suatu mesin k dianggap sebagai mesin kritis sehubungan dengan jadwal S , jika S_k memiliki satu atau beberapa busur yang dilewati lintasan terpanjang dalam D_s . Cara ini ternyata hanya dapat mengelompokkan mesin-mesin sebagai mesin kritis dan non kritis, tanpa membedakan derajat atau tingkatan dari mesin-mesin yang akan menjadi bottleneck.
2. **Bottleneck Quality**, yaitu suatu cara yang dapat memprioritaskan mesin-mesin secara lebih rinci lagi yang akan menjadi mesin bottleneck. Kualitas bottleneck suatu mesin dapat diukur dengan cara ini, misalnya nilai dari marginal utilitynya dalam mengurangi makespan.

Kualitas bottleneck suatu mesin k merupakan hasil penjadwalan optimal satu mesin pada mesin k . Selanjutnya untuk memperjelas konsep ini, akan dijabarkan sebagai berikut :

$M_o \subset M$ adalah himpunan mesin-mesin yang telah terjadwal, dengan memilih $S_p, p \in M_o$ dan untuk setiap mesin $k, k \in M \setminus M_o$, maka kita bentuk problem $(P(k, M_o))$ yang diperoleh dari (P) dengan cara :

- i. mengganti setiap himpunan busur disjunctive $E_p, p \in M_o$ dengan S_p .
- ii. menghapus setiap himpunan busur disjunctive $E_p, p \in M \setminus M_o, p \neq k$.

Permasalahan ini ekuivalen dengan problem penjadwalan satu mesin untuk mesin k guna meminimasi maksimum lateness dengan due date. Kemudian mesin m disebut bottleneck diantara mesin-mesin yang belum terjadwal ($M \setminus M_o$), jika $v(m, M_o) = \max \{v(k, M_o) : k \in M \setminus M_o\}$, dimana $v(k, M_o)$ adalah nilai optimal dari penyelesaian $(P(k, M_o))$. Langkah ini berarti mencari nilai minimasi makespan yang terbesar diantara mesin-mesin yang belum dijadwalkan ($M \setminus M_o$). Nilai optimal minimasi makespan ini diperoleh dari penjadwalan satu mesin dan mesin yang memiliki nilai terbesar adalah mesin bottleneck yang akan dijadwalkan.

Secara umum langkah-langkah dalam algoritma Shifting Bottleneck adalah sebagai berikut :

M_o himpunan mesin-mesin yang telah terjadwal, pada awalnya $M_o = \emptyset$.

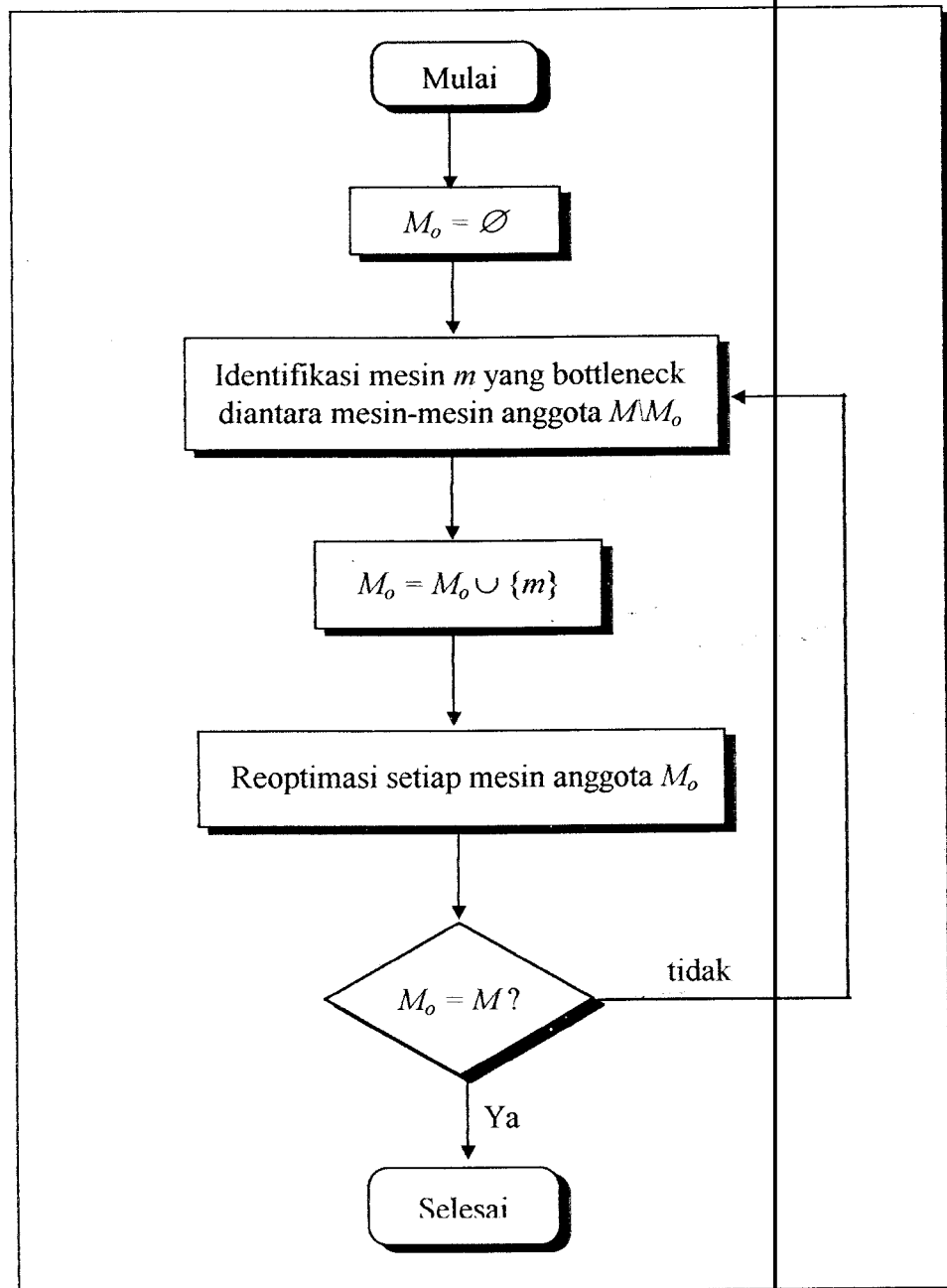
Step 1. Identifikasi mesin bottleneck m diantara mesin-mesin $k, k \in M \setminus M_o$ dan jadwalkan secara optimal.

Set $M_o \leftarrow M_o \cup \{m\}$ dan lanjutkan ke step 2.

Step2. Reoptimasi jadwal dari setiap mesin $k \in M_o$ dengan membuat urutan jadwal yang lain tetap tidak berubah.

Set $M_o' := M_o - \{k\}$ dan selesaikan $P(k, M_o')$.

Jika $M_o = M$ berhenti, jika tidak kembali ke step1.



Gambar 4.2. Flow Chart Algoritma Shifting Bottleneck

4.3. Algoritma Lintasan Terpanjang

Guna mengidentifikasi mesin bottleneck mana yang akan dijadwalkan diantara mesin $k \in M \setminus M_o$ kita selesaikan problem

Minimum t_n

$$t_j - t_i \geq d_i \quad (i,j) \in \cup (S_p : p \in M_o) \cup A$$

$$t_i \geq 0 \quad i \in N$$

$$t_j - t_i \geq d_i \vee t_i - t_j \geq d_j \quad (i,j) \in E_k \quad (P(k, M_o))$$

Juga untuk melakukan reoptimasi urutan dari setiap mesin kritis $k \in M_o$, maka untuk tiap mesin kita selesaikan problem dalam bentuk $(P(k, M_o'))$ dimana $M_o' \subset M_o$.

Permasalahan $(P(k, M_o))$ ekuivalen dengan permasalahan penjadwalan untuk mesin k dengan tujuan untuk meminimasi maksimum lateness dimana setiap operasi yang melalui mesin k memiliki waktu proses d_i , release time r_i , dan due date f_i . Disini $r_i = L(0, i)$ dan $f_i = L(0, n) - L(i, n) + d_i$ dengan $L(i, j)$ merupakan lintasan terpanjang dari i ke j dalam jaringan D_T dan $T := \cup (S_p : p \in M_o)$. Permasalahan ini kemudian dapat dipandang sebagai masalah minimum makespan dimana setiap job harus diproses melalui tiga mesin dengan mesin pertama dan ketiga memiliki kapasitas yang tak terbatas sedangkan mesin kedua yaitu mesin k pada model diatas, memproses satu job setiap saat dan waktu proses dari job i adalah r_i pada mesin pertama, d_i pada mesin kedua dan $q_i := L(0, n) - f_i$ pada mesin ketiga. Harga dari r_i dan q_i sering disebut “head” dan “tail” dari job i . Jadi permasalahan satu mesin yang kita selesaikan dalam algoritma diatas akan berbentuk sebagai berikut :

$$\min t_n$$

$$t_n - t_i \geq d_i + q_i,$$

$$t_i \geq r_i, \quad i \in N^*,$$

$$t_j - t_i \geq d_i \vee t_i - t_j \geq d_j, \quad (i,j) \in E_k, \quad (P^*(k, M_0))$$

dimana r_i dan q_i didefinisikan seperti diatas dan N^* himpunan dari job-job yang diproses pada mesin k .

4.3.1. Algoritma Perhitungan Lintasan Terpanjang

Perhitungan lintasan terpanjang yang digunakan dalam menentukan lintasan terpanjang suatu simpul menggunakan suatu algoritma yang didapatkan dengan memodifikasi dari algoritma route terpendek¹. Algoritma yang telah dimodifikasi ini berusaha mencari simpul-simpul yang berpengaruh terhadap panjang lintasan suatu simpul yang akan dicari. Algoritma ini terus berjalan sampai menemukan simpul akhir atau simpul yang lintasan terpanjangnya telah ditemukan. Selanjutnya, setelah simpul akhir ditemukan, dilakukan penelusuran rekursif dari simpul akhir ke simpul sebelumnya. Nilai lintasan terpanjang dari suatu simpul j adalah maksimum {lintasan terpanjang satu simpul pendahulunya atau simpul i + jarak simpul i ke simpul j }

t_j = jarak terjauh dari simpul awal atau simpul nol (O) ke simpul j

$$t_O = 0$$

d_{ij} = jarak dari simpul i ke simpul j dalam jaringan.

¹Hamdy A Taha., Operations Research, Third Edition, Macmillan Publishing Co Inc. New York, 1982 hal 205

$$t_j = \text{Maksimum}\{t_j + d_{ij}\}, \text{ dimana } i \in N$$

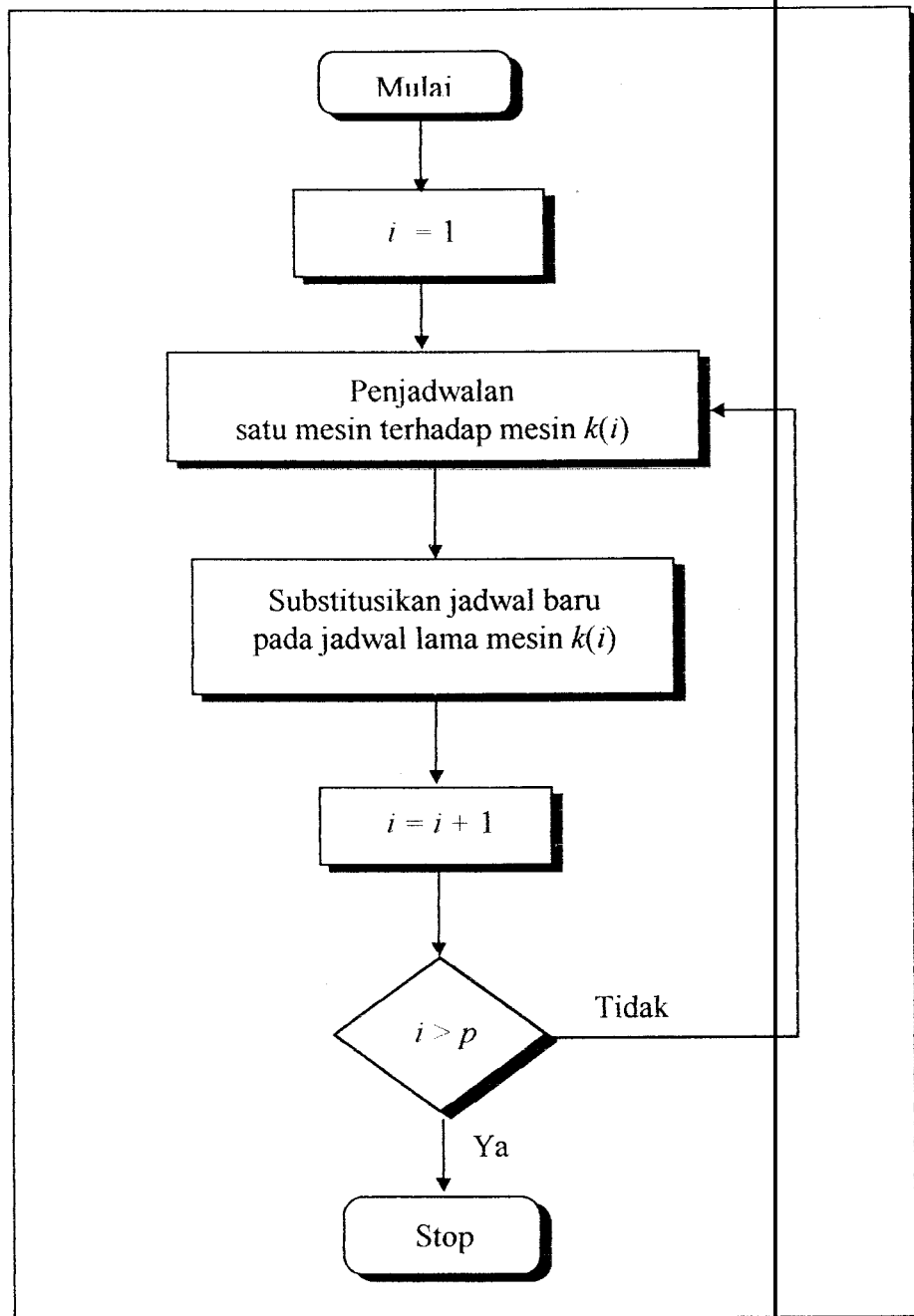
4.4. Prosedur Reoptimasi Lokal

Tujuan dari reoptimasi lokal ini adalah untuk melakukan penjadwalan lagi yang lebih baik terhadap mesin-mesin yang telah terjadwal sebelumnya setelah terdapat anggota mesin baru yang masuk menjadi anggota mesin yang terjadwal.

M_o adalah himpunan mesin-mesin yang telah terjadwal dan $k(1), \dots, k(p)$ merupakan urutan mesin-mesin anggota M_o (disini $p = |M_o|$). Kemudian siklus reoptimasi lokal akan dimulai sebagai berikut :

1. Untuk $i = 1, \dots, p$ selesaikan masalah $(P^*(k(i), M_o \setminus \{k(i)\}))$ dan gantikan pemilihan optimal $S_{k(i)}$ pada jadwal sebelumnya.
2. Sejauh $|M_o| < |M|$ dikerjakan paling banyak tiga kali siklus reoptimasi lokal terhadap setiap himpunan M_o .
3. Pada langkah terakhir ketika $|M_o| = |M|$, kita lakukan reoptimasi lokal sampai titik dimana tidak ada perbaikan untuk suatu siklus penuh.

Masalah $(P^*(k(i), M_o \setminus \{k(i)\}))$ yang ditemui dalam reoptimasi lokal dapat diselesaikan dengan cara yang sama seperti masalah $(P^*(k, M_o))$. Urutan $k(1), \dots, k(p)$ dari M_o pada awalnya merupakan urutannya masuk menjadi anggota M_o . Setiap kali satu siklus penuh selesai, anggota dari M_o diurutkan lagi berdasarkan menurunnya nilai solusi pada masalah $(P^*(k(i), M_o \setminus \{k(i)\}))$.



Gambar 4.3. Flow Chart Reoptimasi Lokal

4.5. Penjadwalan Satu Mesin

Penjadwalan satu mesin ini merupakan dasar dari algoritma Shifting Bottleneck, baik untuk identifikasi mesin bottleneck dan reoptimasi lokal. Algoritma penjadwalan satu mesin ini² digunakan untuk menjadwalkan mesin satu per satu secara optimal.

M_1 dan M_3 adalah mesin non bottleneck yang mempunyai kapasitas tak terhingga sedangkan M_2 adalah mesin bottleneck yang hanya memproses satu operasi setiap saat. Terdapat n operasi yang harus diproses pada mesin M_1 , M_2 , M_3 (dengan urutan demikian) dan operasi i memerlukan r_i di M_1 , d_i di M_2 dan q_i di M_3 . Tujuan dari penjadwalan ini adalah meminimasi makespan.

Masalah penjadwalan satu mesin ini akan dinyatakan dalam bentuk conjunctive graph yang dinotasikan $G = (X, U)$, dimana :

- ♦ X merupakan himpunan operasi-operasi yang digambarkan dalam bentuk simpul-simpul yang diperoleh dengan menambahkan dua simpul fiktif, yaitu O dan $*$ pada himpunan operasi I .

$X = I \cup \{O, *\}$, O menyatakan simpul mulai atau awal operasi dan $*$ adalah akhir operasi.

- ♦ U merupakan gabungan dari tiga himpunan busur-busur, $U = U_1 \cup U_2 \cup U_3$ dimana

$U_1 = \{(O, i) \mid i \in I\}$; busur (O, i) bernilai r_i dan ini berarti operasi i tidak dapat dimulai sebelum waktu r_i .

²Jacques Carlier., The one-machine sequencing problem, European Journal of Operation Research 11 (1982) 42-47.

$U_2 = \{(i,*) \mid i \in I\}$; busur $(i,*)$ bernilai $q_i + d_i$, hal ini karena operasi i masih memerlukan waktu $q_i + d_i$ di dalam sistem sesaat setelah waktu mulai operasinya.

$U_3 = \{(i,j) \mid \text{operasi } i \text{ mendahului operasi } j\}$; busur (i,j) bernilai d_i .

Jadwal yang dihasilkan $J = \{t_i \mid i \in X\}$ merupakan kumpulan waktu mulai t_i dari operasi i yang besarnya sama dengan lintasan terpanjang dari O ke i dalam conjunctive graph G , sedangkan t_o bernilai nol dan t_* sama dengan nilai critical path (lintasan kritis) dalam hal ini nilai dari makespannya. Tujuan dari algoritma ini adalah untuk mendapatkan suatu jadwal dengan urutan tertentu yang dapat meminimasi nilai dari lintasan kritisnya (makespan) dalam conjunctive graph.

Proposisi 1:

Untuk semua $I_I \subseteq I$

$$h(I_I) = \text{Min } r_i + \sum d_i + \text{Min } q_i, \text{ untuk } i \in I_I$$

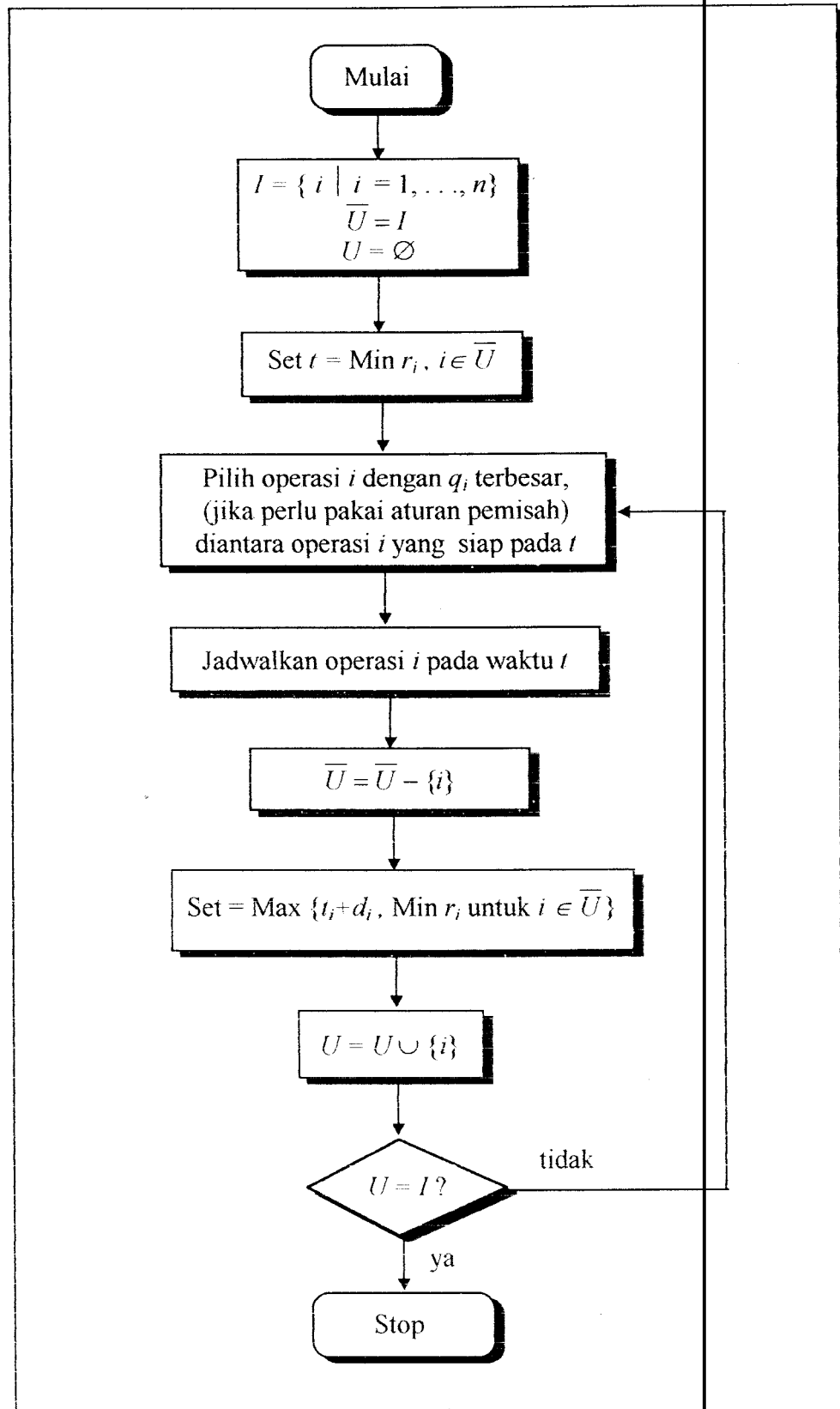
adalah lower bound dari makespan optimal.

Selanjutnya dalam masalah penjadwalan satu mesin ini kita memakai algoritma Schrage. Dalam algoritma Schrage ini, operasi yang siap dengan q_i terbesar dijadwalkan terlebih dahulu. Secara rinci akan dijelaskan sebagai berikut :

U adalah himpunan operasi-operasi yang telah dijadwalkan, sedangkan \overline{U} adalah himpunan dari sisa operasi-operasi yang belum dijadwalkan. I adalah himpunan operasi-operasi yang akan dijadwalkan dan t adalah waktu.

Selanjutnya langkah-langkah dalam algoritma Schrage adalah sebagai berikut :

- 1) Set $t = \text{Min } r_i, i \in I; U = \emptyset$
- 2) Pada waktu t , jadwalkan diantara operasi-operasi yang siap ($r_i \leq t$) dari \overline{U} , pilih operasi i yang memiliki q_i terbesar, jika ada lebih dari satu operasi yang sama, pilih operasi i yang memiliki d_i terbesar dan jika masih ada lebih dari satu operasi yang tetap sama, maka pilih salah satu secara random.
- 3) Set $U = U \cup \{i\}; t_i = t; t = \text{Max}(t_i + d_i, \text{Min } r_i \text{ untuk } i \in \overline{U})$.
Jika $U = I$, maka berhenti, jika tidak lanjutkan ke 2).



Gambar 4.4. Flow Chart Algoritma Schrage

Teorema

L merupakan makespan yang dihasilkan oleh algoritma Schrage.

- a) Jika jadwal yang dihasilkan tidak optimal, maka terdapat sebuah operasi kritis c dan himpunan kritis J sehingga :

$$h(J) = \text{Min } r_i + \sum d_i + \text{Min } q_i > L - d_c ; i \in J.$$

Akibatnya, selisih makespan antara jadwal optimal dengan jadwal yang dihasilkan algoritma Schrage kurang dari d_c dan pada jadwal yang optimal operasi c akan diproses sebelum atau sesudah operasi-operasi anggota himpunan J .

- b) Jika jadwal ini optimal, maka terdapat himpunan J sehingga $h(J) = L$.

Definisi dari J dan c

Operasi-operasi yang dilalui lintasan kritis dari simpul nol sampai dengan simpul akhir disebut operasi-operasi kritis. Kemudian dari operasi-operasi kritis itu, kita modifikasi nomor simpulnya menjadi $0, 1, \dots, p, *$ dan p disini merupakan operasi terakhir yang dilalui lintasan kritis. Dengan demikian nilai lintasan kritis tersebut adalah $L = r_1 + \sum d_i + q_p$; dimana $i = 1, \dots, p$.

Jika terdapat $q_p = \text{Min } q_i$, maka nilai dari lintasan kritisnya menjadi

$$L = r_1 + \sum_{i=1, \dots, p} d_i + q_p$$

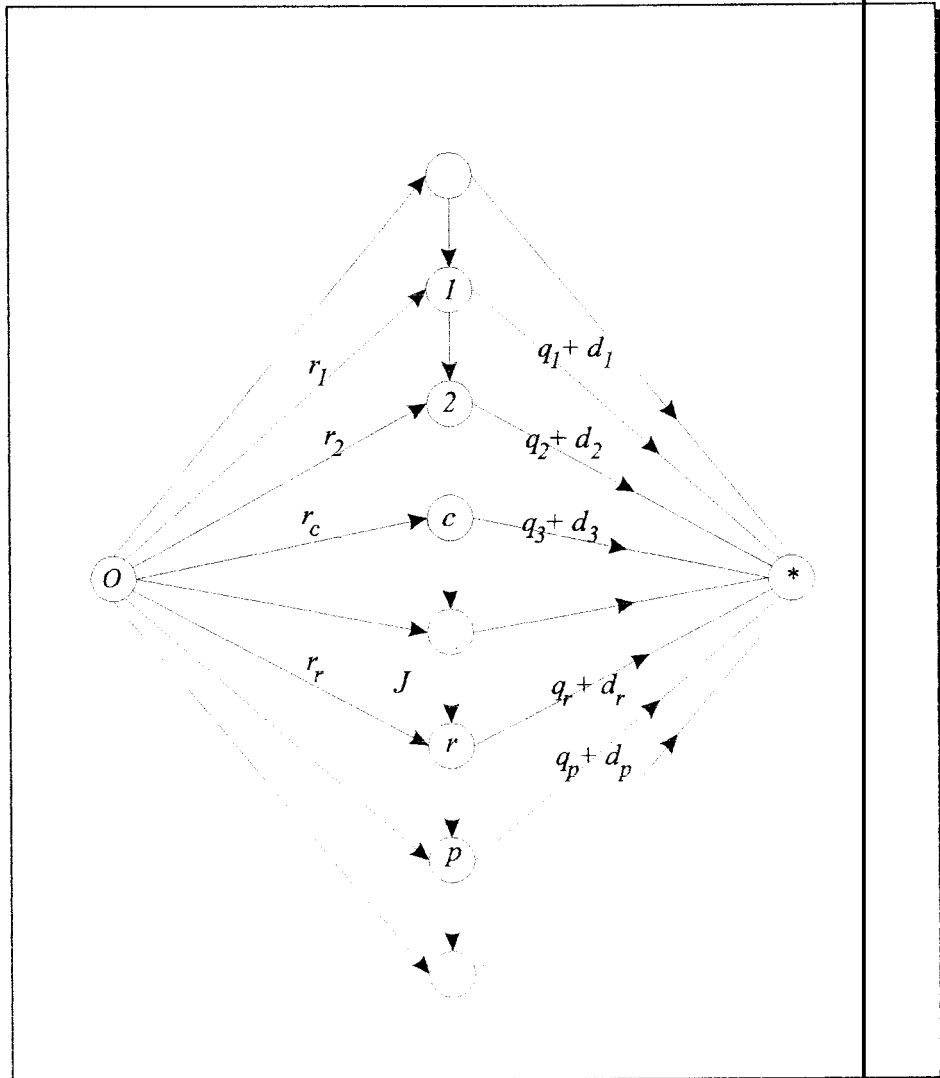
$$L = \text{Min}_{i=1, \dots, p} r_i + \sum_{i=1, \dots, p} d_i + \text{Min}_{i=1, \dots, p} q_i$$

Sehingga $L = h(J)$ dengan $J = \{1, 2, \dots, p\}$ dan jadwal ini merupakan jadwal Schrage yang optimal. Apabila terdapat $i < p$ sehingga $q_i < q_p$, maka akan

terdapat suatu operasi kritis c yang terdekat operasi kritis p sehingga $q_c < q_p$.

Kemudian dibentuk himpunan $J = \{c+1, \dots, p\}$ sehingga $q_c < q_r$ untuk $r \in J$.

Pada jadwal yang optimal, operasi kritis c akan diproses sebelum atau sesudah operasi-operasi anggota himpunan J . Untuk lebih jelasnya dapat dilihat pada gambar 4.5. dibawah ini



Gambar 4.5. Pendefinisian J dan c

Metoda Branch and Bound

Metoda Branch and Bound yang dipakai dalam penjadwalan satu mesin ini berdasarkan algoritma Schrage, himpunan kritis J dan operasi kritis c .

Deskripsi Pohon

Pohon disini merupakan rangkaian dari simpul-simpul dan setiap simpul (S) yang ada merupakan urutan jadwal yang dibentuk dari penjadwalan satu mesin dengan lower bound $f(S)$ dan upper bound f_0 adalah solusi terbaik yang telah diketahui. Jadi pohon merupakan konfigurasi jadwal yang dapat dibentuk dari penjadwalan satu mesin.

Pencabangan (Branching)

Cabang yang diperhatikan dari pohon adalah cabang yang memiliki simpul dengan lower bound yang terkecil. Kemudian dari simpul tersebut diterapkan algoritma Schrage.

Jika operasi kritis c tidak ada, maka secara teoritis jadwal tersebut optimal, akan tetapi jika terdapat operasi kritis c , maka operasi c akan diproses sebelum atau sesudah operasi-operasi anggota himpunan J .

Dua masalah akan muncul setelah ditemukannya operasi kritis c , yaitu :

1. Operasi kritis c diproses sebelum operasi-operasi himpunan J dengan membuat

$$q_c = \text{Max} \left(q_c, \sum_{r \in J} d_r + q_p \right)$$

kecil dari f_0 , maka upper bound sekarang yaitu f_0 sama dengan makespan dari jadwal tersebut.

4.6. Pengembangan Model Dasar

Pada bagian ini akan dicoba untuk mengembangkan model dasar yang telah dijelaskan pada bagian sebelumnya untuk menangani masalah penjadwalan job shop yang lebih kompleks dengan kriteria yang diukur masih tetap minimasi makespan.

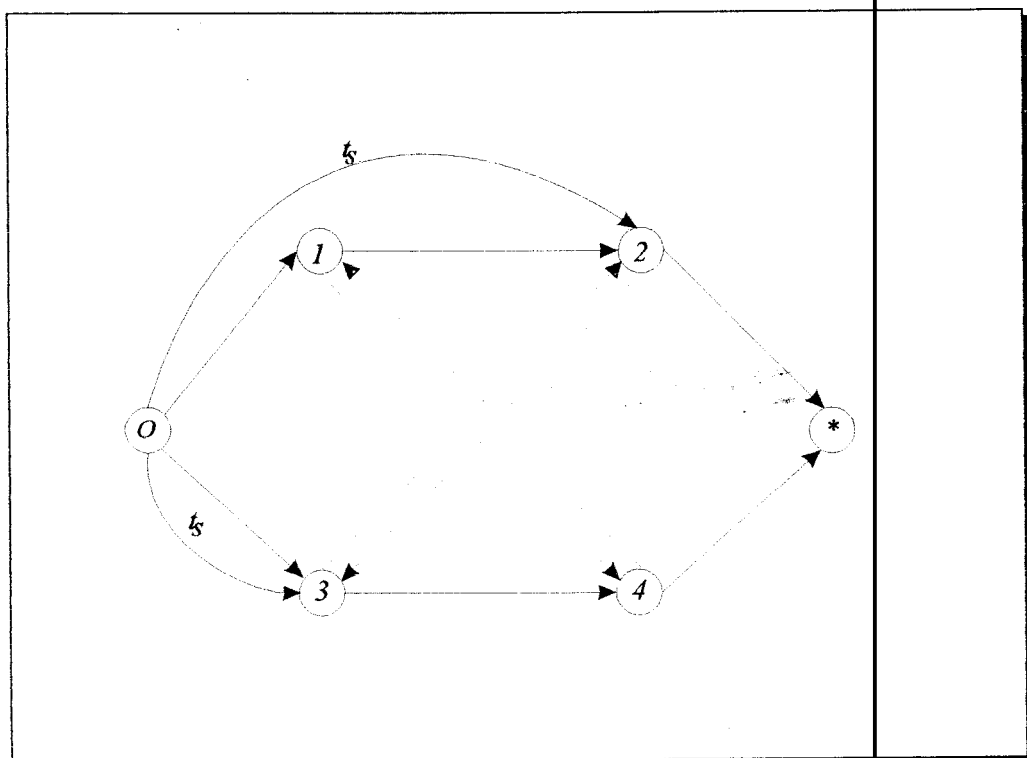
4.6.1. Waktu Siap Job yang tidak sama dengan nol

Kebanyakan model atau algoritma yang selama ini digunakan untuk menyelesaikan masalah penjadwalan job shop, mengasumsikan bahwa semua job pada saat akan dijadwalkan siap pada waktu $t = 0$ atau dengan kata lain bahwa job-job yang akan dijadwalkan semuanya tersedia pada saat awal proses penjadwalan dilakukan. Model ini tidak mampu untuk mengatasi adanya job-job yang tidak siap pada $t = 0$ (pada saat job-job akan dijadwalkan). Hal ini bisa saja terjadi mungkin karena ada bahan baku atau material yang diperlukan belum tersedia. Keadaan seperti juga tidak diperhatikan dalam model dasar. Oleh karena itu akan dicoba untuk mengembangkan model dasar guna mengatasi masalah tersebut. Pengembangan model dasar ini mengasumsikan bahwa waktu siap dari job yang tidak siap pada waktu $t = 0$ telah diketahui. Pengembangan ini dilakukan dengan cara menambahkan suatu busur dari simpul awal (O) ke

4.6.2. Waktu Siap mesin yang tidak sama dengan nol

Terkadang ada beberapa mesin yang digunakan dalam penjadwalan tidak siap pada $t = 0$, hal ini bisa terjadi karena mungkin operator yang menangani mesin tersebut belum siap atau mesin sedang diperbaiki. Untuk mengatasi masalah seperti ini, kita kembangkan model dasar seperti model untuk waktu siap job yang tidak sama dengan nol, hanya saja penambahan busur diberikan kepada simpul-simpul atau operasi-operasi yang dikerjakan pada mesin yang tidak siap tersebut.

Misal operasi 2 dan 3 dikerjakan pada mesin 2, mesin 2 siap pada $t = t_s$



Gambar 4.8. Mesin 2 siap pada $t = t_s$

4.6.3. Adanya Job-job yang berprioritas.

Pengembangan algoritma untuk job-job yang berprioritas tetap didasarkan pada penjadwalan satu mesin dengan algoritma Carlier. Algoritma ini mendahulukan operasi dengan prioritas yang lebih tinggi jika terdapat lebih dari satu operasi yang siap.

Selanjutnya algoritma Schrage sedikit mengalami perubahan, yaitu sebagai berikut :

- 1) Set $t = \min r_i, i \in I; U = \emptyset$
- 2) Pada waktu t , jadwalkan diantara operasi-operasi yang siap ($r_i \leq t$) dari \bar{U} , pilih operasi i yang memiliki prioritas terbesar, jika ada lebih dari satu operasi yang sama, pilih operasi i yang memiliki q_i terbesar dan jika masih ada lebih dari satu operasi yang tetap sama, maka pilih salah satu secara random.
- 3) Set $U = U \cup \{i\}; t_i = t; t = \max(t_i + d_i, \min r_i \text{ untuk } i \in \bar{U})$.

Jika $U = I$, maka berhenti, jika tidak lanjutkan ke 2).

Selanjutnya untuk prosedur pencabangan dimana terdapat operasi c , maka pencabangan dimana operasi c diproses setelah operasi-operasi anggota himpunan J , akan dilakukan hanya jika operasi-operasi anggota himpunan J memiliki prioritas yang sama atau lebih tinggi dari operasi c .

4.6.4. Adanya Routing Kompleks

Routing kompleks dari suatu job merupakan urutan permesinan dari job tersebut yang diroses lebih dari satu kali pada mesin yang sama, contohnya adalah sebagai berikut :

		Operasi			
		1	2	3	4
Job	1	1	2	3	4
	2	2	1	2	3
	3	3	4	1	2

Gambar 4.9. Routing Kompleks

Masalah seperti ini dapat diselesaikan dengan algoritma Shifthing Bottleneck sepanjang tidak terjadi kesalahan pengurutan operasi dalam penjadwalan satu mesin untuk operasi-operasi job tersebut. Jika terjadi kesalahan pengurutan operasi, maka akan terjadi siklus dalam graph yang terjadi. Hal ini menyebabkan iterasi tak terhingga pada perhitungan lintasan terpanjang.

Oleh karena itu algoritma dasar perlu dikembangkan dengan cara melakukan perubahan terhadap penjadwalan satu mesin. Perubahan ini dilakukan untuk menjamin bahwa operasi-operasi yang terlibat dalam penjadwalan satu mesin masih tetap mengikuti precedence constraint. Perubahan ini dilakukan pada teknik pencabangan branch and bound. pada saat terjadi

perubahan konfigurasi dimana operasi c akan diproses sebelum atau sesudah operasi-operasi anggota himpunan J .

Pencabangan terhadap operasi c yang akan diproses setelah operasi-operasi anggota himpunan J , dilakukan hanya jika operasi-operasi anggota himpunan J tidak mempunyai anggota yang merupakan operasi yang harus diproses setelah operasi c sesuai dengan precedence constraint.

BAB V

PENGUMPULAN DAN PENGOLAHAN DATA

5.1. Aliran Proses Produksi

Pada satu periode tertentu PT BBI mendapat job atau pesanan baik dari luar perusahaan maupun dari pihak PT BBI sendiri untuk mengerjakan proses permesinan beberapa produk. Dibawah ini terdapat daftar nama job atau produk yang memerlukan proses permesinan di PT BBI beserta jumlah pesanan yang harus dipenuhi.

No	Nama Job (Produk)	Jumlah Pesanan
1	Intake Mazda 626	60
2	Intake Mazda 323	80
3	Exhaust Mazda 323	80
4	Exhaust Mazda MR 90	180
5	Crank Case Yanmar TS 230	50
6	Crank Case Yanmar TS 300	40
7	Counter Weight F2L-912	100
8	Counter Weight F3L-912	120
9	Counter Weight F4L-912	90
10	Counter Weight F6L-912	85
11	Cylinder Head FL-913	46
12	Cylinder Head FL-912	125
13	Bearing Cap F2L-912	127
14	Trash Bearing F2L-912	100

Sebelum semua job dikerjakan, terlebih dahulu dilakukan kegiatan administrasi dan persiapan-persiapan selama satu hari dimana dalam satu hari terdapat delapan jam kerja.

Untuk job pertama sampai job keenam dilakukan pengepakan atau pemetian sedangkan job lainnya tidak dilakukan pengepakan. Di bawah ini terdapat daftar yang menunjukkan kapasitas peti dan waktu pemetian.

No	Nama Job	Kapasitas Peti	Waktu
1	Intake Mazda 626	20 buah	1 jam 20 menit
2	Intake Mazda 323	20 buah	4 jam
3	Exhaust Mazda 323	20 buah	4 jam
4	Exhaust Mazda MR 90	60 buah	4 jam
5	Crank Case Yanmar TS 230	10 buah	1 jam
6	Crank Case Yanmar TF 300	10 buah	1 jam

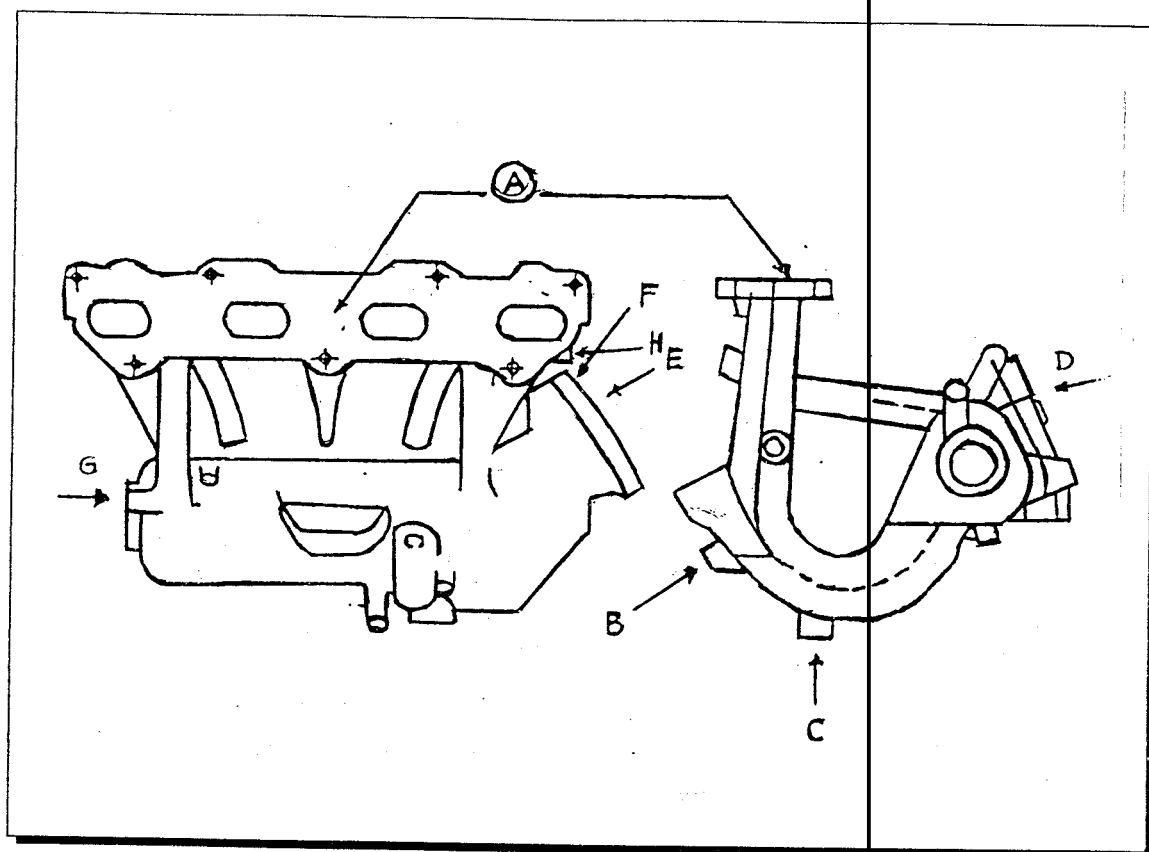
Ukuran peti untuk job-job diatas berbeda-beda, kecuali ukuran peti untuk Crank Case. Pengepakan untuk Intake Mazda 323 dan Exhaust Mazda 323 dilakukan dalam satu peti ditambah dengan 20 buah penutup (cover). Cover tersebut sudah disediakan oleh pihak pemesan sebagai pasangan Intake Mazda 323 dan Exhaust Mazda 323.

Masing-masing job memerlukan proses permesinan atau urutan pengerjaan yang berbeda-beda. Waktu permesinan atau machining time sudah ditetapkan oleh pihak perusahaan, baik waktu permesinan untuk mesin-mesin NC maupun mesin-mesin yang dioperasikan secara manual. Sedangkan untuk waktu loading-unloading

dan waktu untuk pergantiann fixture belum ditentukan oleh pihak perusahaan, sehingga perlu dilakukan pengukuran kerja.

Selanjutnya akan dibahas proses produksi atau proses permesinan untuk masing-masing job atau produk.

1. Intake Mazda 626

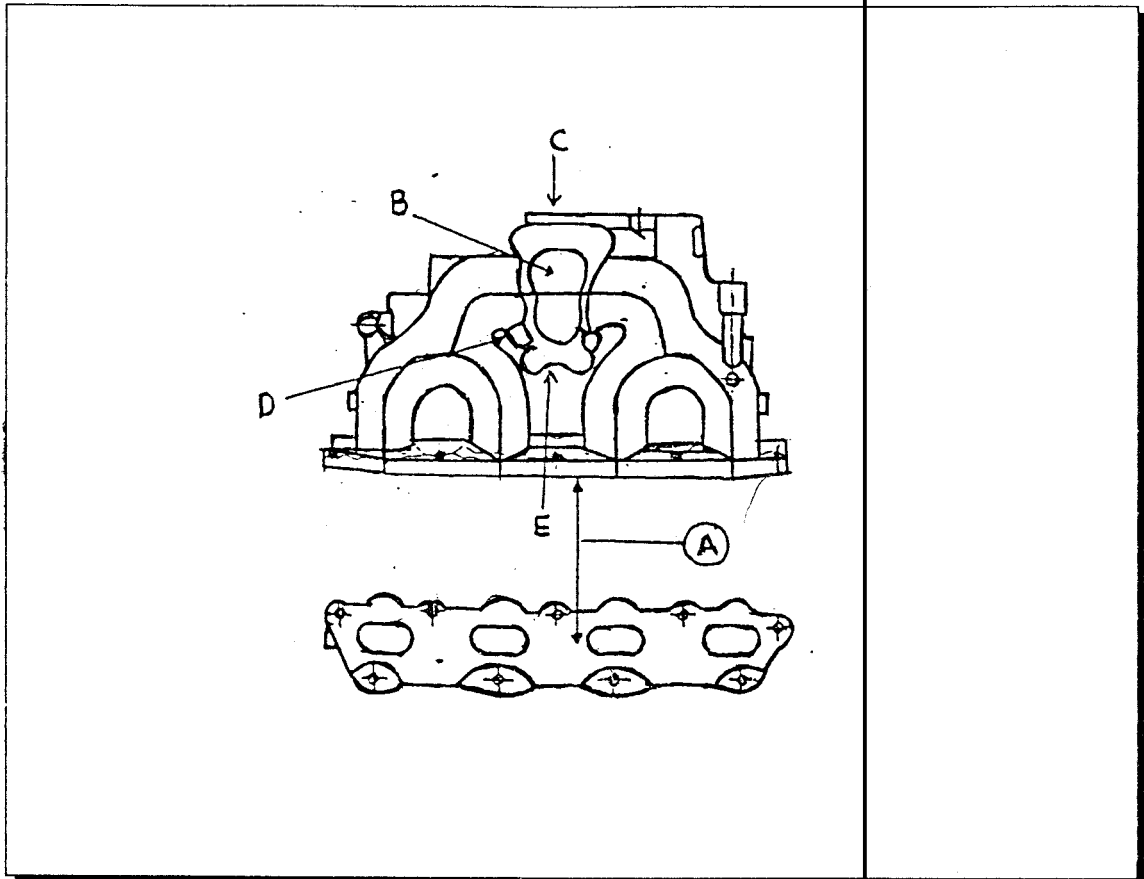


Gambar 5.1.
Intake Mazda 626

Proses	Mesin	Waktu Proses	Keterangan
1	VMC. 45 [43]	10 menit	Permukaan A
2	BMC. 40 [52]	20 menit	Permukaan B, C, D
3	VMC. 45 [43]	15 menit	Permukaan E, 2 palet

4	VMC. 45 [43]	15 menit	Permukaan F
5	Radial Drill [35]	10 menit	Pengeboran G, H
6	Radial Drill [35]	6 menit	
7	Deboring	5 menit	
8	Jet Washer [72]	10 menit	15 part

2. Intake Mazda 323

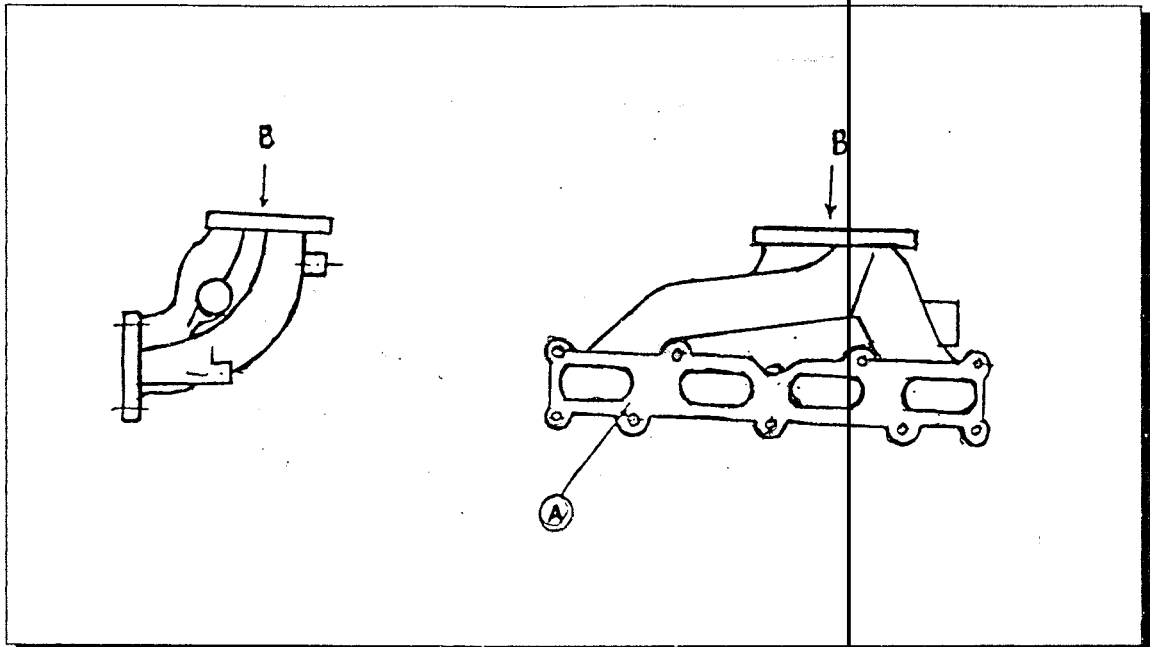


Gambar 5.2.
Inatek Mazda 323

Proses	Mesin	Waktu Proses	Keterangan
1	BMC. 40 [51]	10 menit	Permukaan A
2	BMC. 40 [51]	20 menit	Permukaan B
3	VMC. 45 [43]	20 menit	Permukaan C

4	Radial Drill [34]	5 menit	Pengboran D, E
5	Radial Drill [34]	5 menit	
6	Deboring	5 menit	
7	Jet Washer [72]	10 menit	15 part

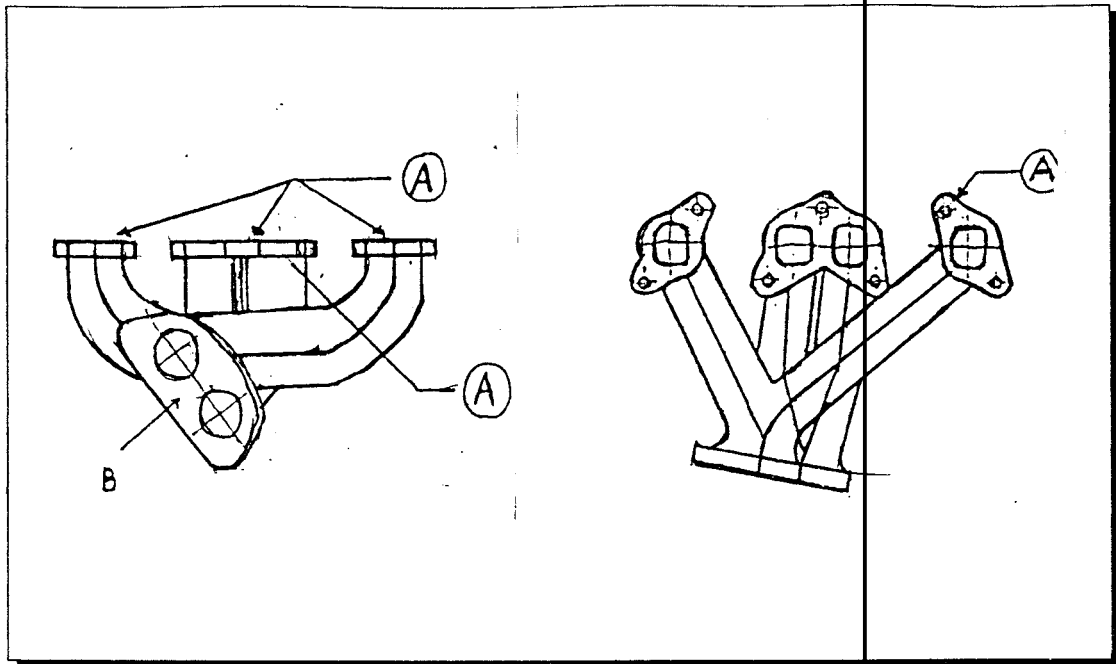
3. Exhaust Mazda 323



Gambar 5.3.
Exhaust Mazda 323

Proses	Mesin	Waktu Proses	Keterangan
1	BMC. 40 [53]	15 menit	Permukaan A. 2 palet
2	BMC. 40 [53]	10 menit	Permukaan B
3	Radial Drill [34]	5 menit	
4	Deboring	2 menit	
5	Jet Washer [72] dan Lubricating	10 menit	15 part

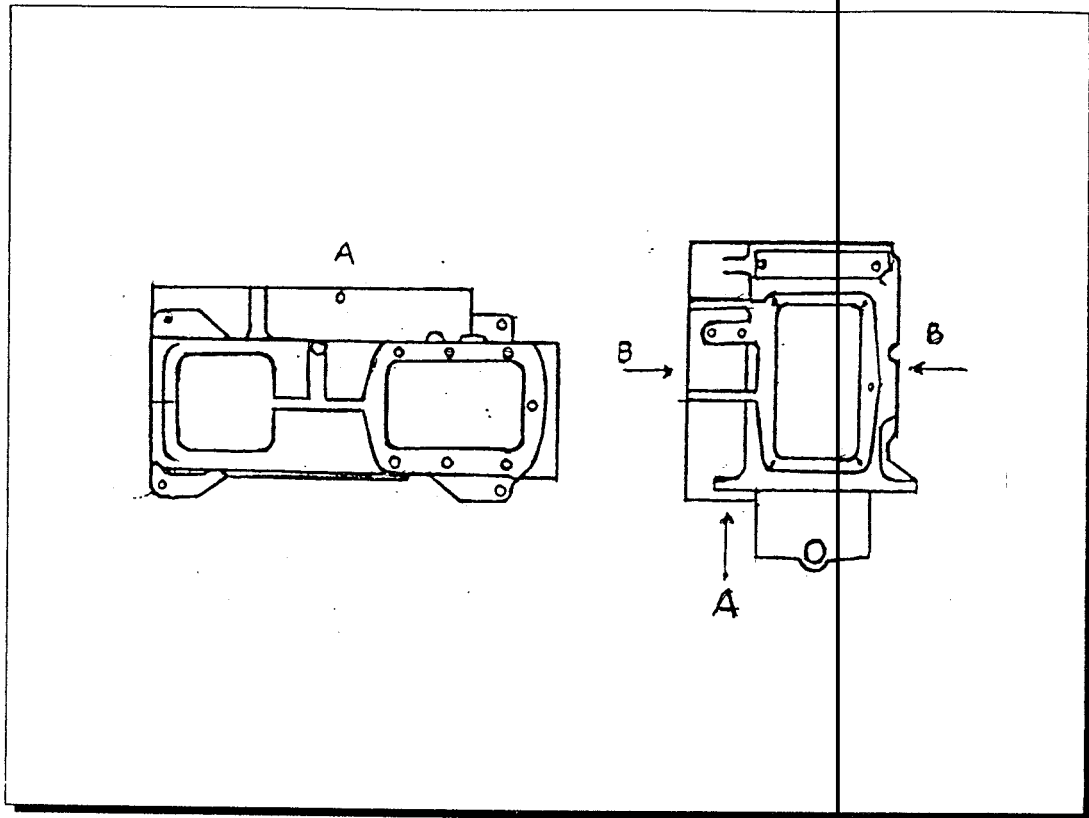
4. Exhaust Mazda MR 90



Gambar 5.4.
Exhaust Mazda MR 90

Proses	Mesin	Waktu Proses	Keterangan
1	VMC. 45 [41]	20 menit	Permukaan A, 2 part
2	NC Horizontal Boring [150]	20 menit	Permukaan B, 2 part
3	Radial Drill [34]	1 menit	
4	Deboring	1,5 menit	
5	Jet Washer [72] dan Lubricating	10 menit	15 part

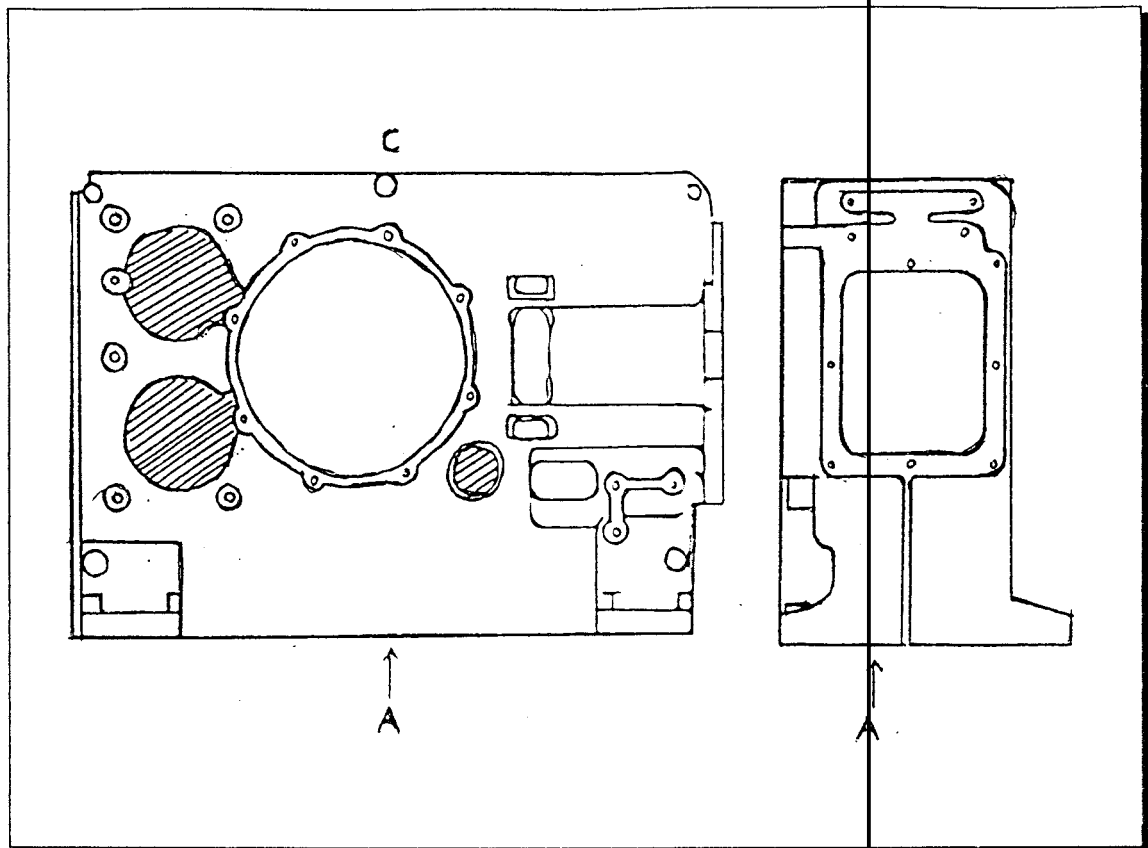
5. Crank Case Yanmar TS 230



Gambar 5.5.
Crank Case Yanmar TS 230

Proses	Mesin	Waktu Proses	Keterangan
1	BMC. 80 [13]	31 menit	Permukaan A
2	BMC. 80 [16]	46 menit	Permukaan B
3	BMC. 80 [17]	43 menit	Permukaan B
4	BMC. 80 [12]	46 menit	Pengeboran
5	Jet Washer [72] dan Lubricating	32 menit	7 part

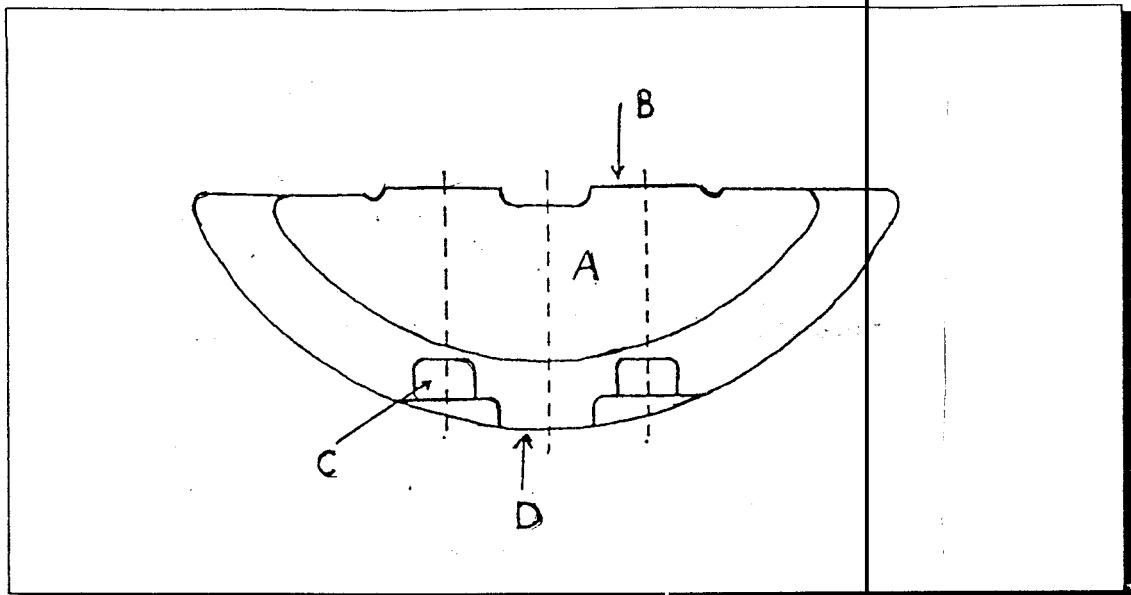
6. Crank Case Yanmar TF 300



Gambar 5.6.
Crank Case Yanmar TF 300

Proses	Mesin	Waktu Proses	Keterangan
1	BMC. 80 [14]	32 menit	Permukaan A
2	BMC. 80 [15]	51 menit	Permukaan C
3	BMC. 80 [18]	52 menit	Permukaan A
4	BMC. 80 [11]	47 menit	Pengeboran
5	Jet Washer [72] dan Lubricating	31 menit	6 part

7. Counter Weight Type F2L-912



Gambar 5.7.
Counter Weight

Proses	Mesin	Waktu Proses	Keterangan
1	Rotary Surface Grinding [410]	32 menit	Permukaan A, 24 part
2	Vertical Milling [420]	4,2 menit	Permukaan B
3	Radial Drill [430]	8,2 menit	Pengeboran C
4	Engine Lathe [070]	2,6 menit	Permukaan D
5	Work Bench [440]	2 menit	Pembersihan

8. Counter Weight Type F3L-912

Proses	Mesin	Waktu Proses	Keterangan
1	Rotary Surface Grinding [410]	35 menit	Permukaan A, 24 part
2	Vertical Milling [420]	4,4 menit	Permukaan B
3	Radial Drill [430]	8,2 menit	Pengeboran C
4	Work Bench [440]	2 menit	Pembersihan

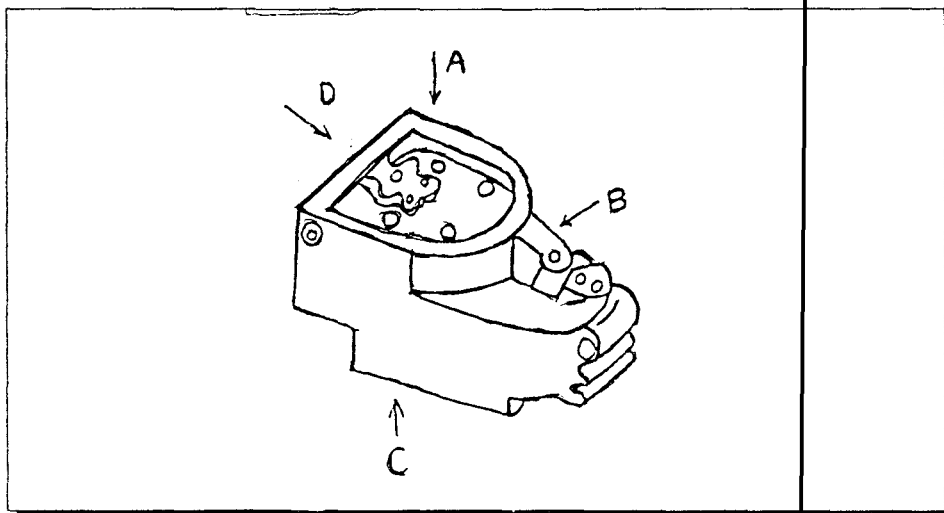
9. Counter Weight Type F4L-912

Proses	Mesin	Waktu Proses	Keterangan
1	Rotary Surface Grinding [410]	33,7 menit	Permukaan A, 24 part
2	Vertical Milling [420]	4,7 menit	Permukaan B
3	Radial Drill [430]	8,2 menit	Pengeboran C
4	Work Bench [440]	2 menit	Pembersihan

10. Counter Weight Type F6L-912

Proses	Mesin	Waktu Proses	Keterangan
1	Rotary Surface Grinding [410]	31,6 menit	Permukaan A, 24 part
2	Vertical Milling [420]	4,2 menit	Permukaan B
3	Radial Drill [430]	8,2 menit	Pengeboran C
4	Work Bench [440]	2 menit	Pembersihan

11. Cylinder Head Type FL-913



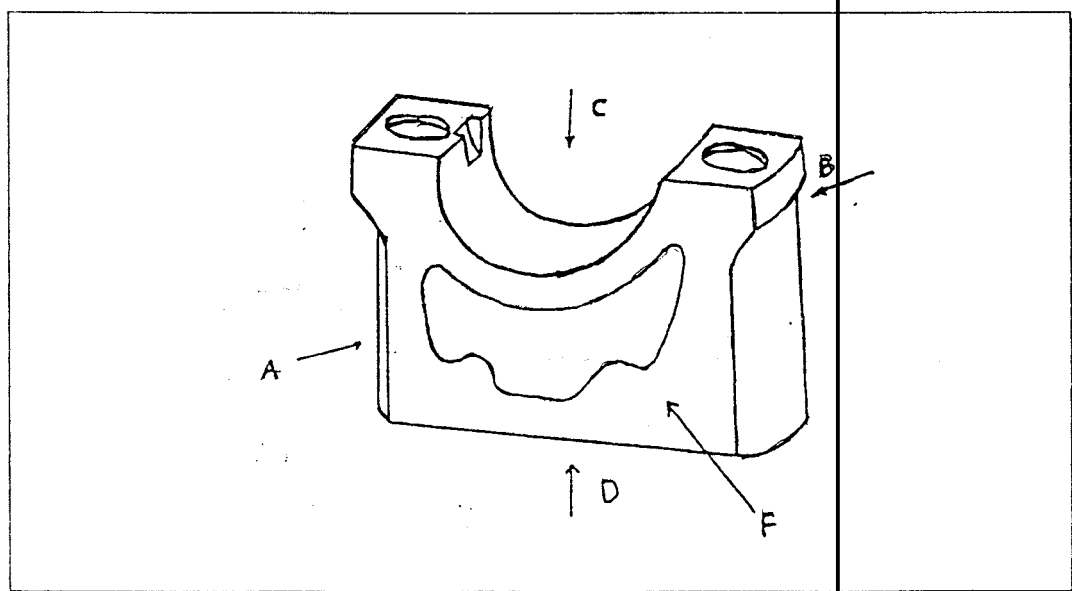
Gambar 5.8.
Cylinder Head

Proses	Mesin	Waktu Proses	Keterangan
1	BMC.40 [51]	24 menit	Permukaan A. B. C. D
2	NC Lathe [62]	2,28 menit	Permukaan A
3	BMC.40 [51]	10,5 menit	Pengeboran
4	Jet Washer [72]	20 mnit	25 part

12. Cylinder Head Type FL-912

Proses	Mesin	Waktu Proses	Keterangan
1	BMC.40 [51]	24 menit	Permukaan A. B. C. D
2	NC Lathe [62]	2,28 menit	Permukaan A
3	BMC.40 [51]	10,5 menit	Pengeboran
4	Jet Washer [72]	20 menit	25 part

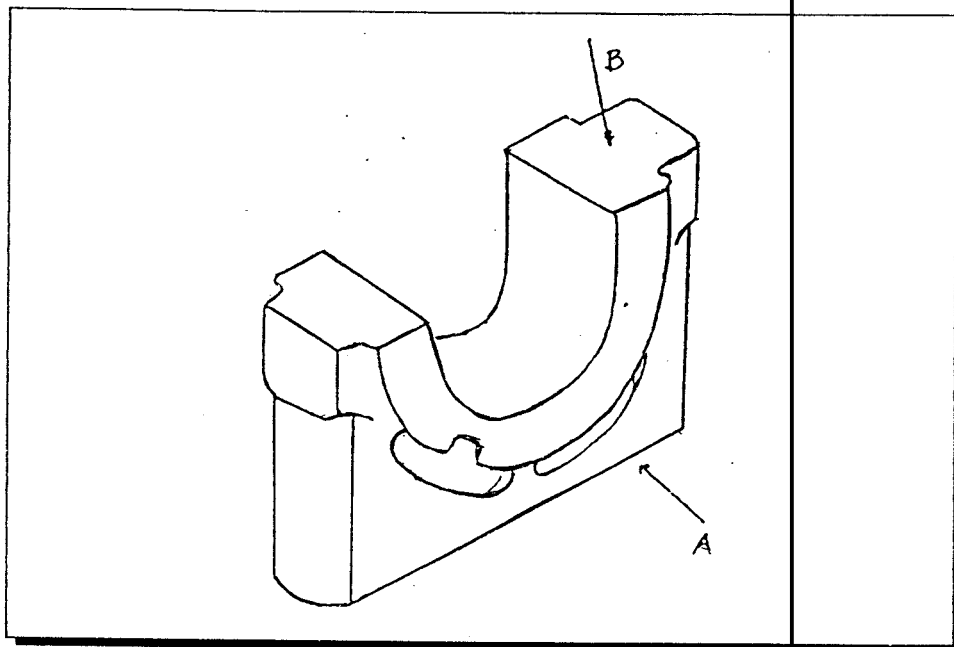
13. Bearing Cap Type F2L-912



Gambar 5.9
Bearing Cap

Proses	Mesin	Waktu Proses	Keterangan
1	Rotary Surface Grinding [410]	15 menit	Permukaan F, 24 part
2	VMC.45 [42]	59,76 mneit	Permukaan A, B, C, D 3 part
3	Jet Washer [72] dan Lubricating	25 menit	100 part

14. Trash Bearing Type F2L-912



Gambar 5.10
Trash Bearing

Proses	Mesin	Waktu Proses	Keterangan
1	Rotary Surface Grinding [410]	30 menit	Permukaan A, 24 part
2	VMC.45 [42]	25 menit	Permukaan B, 3 part
3	Jet Washer [72] dan Lubricating	25 menit	100 part

4.2. Waktu Baku Proses

Untuk mengetahui waktu baku tiap proses permesinan, terlebih dahulu dilakukan pengukuran kerja untuk waktu loading-unloading tiap proses dan waktu pergantian fixture untuk mesin-mesin yang digunakan. waktu loading-unloading adalah waktu yang digunakan oleh operator untuk memasang dan membongkar benda kerja pada fixture termasuk didalamnya waktu pengambilan benda kerja dari tempatnya dan waktu pengambilan benda kerja dari fixture. waktu pergantian fixture adalah waktu yang digunakan oleh operator untuk mengganti fixture pada mesin karena mesin akan digunakan untuk operasi dari job yang berlainan.

Untuk mendapatkan waktu baku tiap operasi, maka yang perlu diketahui adalah waktu permesinan tiap operasi dan waktu loading-unloadingnya. Hal ini karena waktu baku tiap operasi sama dengan waktu permesinan ditambah dengan waktu loading-unloadingnya. Karena waktu loading-unloading tiap operasi belum ditetapkan oleh perusahaan, maka perlu dilakukan pengukuran waktu kerjanya. Untuk operasi deboring dan work bench tidak memerlukan waktu loading-unloading.

Selanjutnya terdapat mesin-mesin yang mempunyai waktu pergantian fixture, karena mesin-mesin ini digunakan untuk mengerjakan lebih dari satu operasi job. Mesin-mesin itu adalah :

No	Mesin	Dari operasi ke operasi
1	BMC. 45 [42]	Bearing Cap ke Trash Bearing
2	Verical Milling [420]	Setiap type Counter Weight
3	Radial Drill [430]	Type Counter Weight
4	VMC. 45 [43]	Intake 626 ke Intake 323
5	Radial Drill [34]	Intake 323 ke Exhaust 323
6	VMC. 45 [43]	Proses 1 ke 3 Intake 626
7	BMC. 40 [51]	Proses 1 ke 2 Intake 323
8	BMC. 40 [51]	Proses 1 ke 3 Cylinder Head 913
9	BMC. 40 [51]	Proses 1 ke 3 Cylinder Head 912

Pada lampiran B dapat dilihat tabel-tabel yang menunjukkan banyaknya pengamatan dan besarnya waktu loading-unloading untuk tiap operasi dan besarnya waktu pergantian fixture untuk tiap mesin. Selanjutnya dilakukan uji kecukupan data, keseragaman data dan uji distribusi.

Hasil pengujian ditabelkan pada tabel-tabel berikutnya sedangkan histogram frekuensi dan pengujian distribusi normal dengan uji Kolmogorov-Smirnov dapat dilihat pada lampiran A.

Keterangan notasi yang digunakan :

N = Banyaknya pengamatan

N^* = Banyaknya pengamatan seharusnya

SD = Standard Deviasi

BKA = Batas Kontrol Atas

BKB = Batas Kontrol Bawah

$$D_n = \sup_x |S(x) - F_0(x)|$$

SL = Significant Level (tingkat kepercayaan).

\bar{X} = Rata-rata pengamatan

Wn = Waktu normal

Ws = Waktu standard atau waktu baku

Performance Rating dilakukan oleh pengamat dimana mempunyai nilai antara 90% sampai 100%. Allowance yang diberikan oleh perusahaan sebesar 13%.

Harga Dn tabel untuk uji dua sisi $p = 0,95$ adalah sebagai berikut :

n	Dn
5	0,563
6	0,519
7	0,483
8	0,454

Dari tabel-tabel hasil perhitungan data pada lampiran B dapat diketahui bahwa banyaknya pengamatan kerja lebih besar dari banyaknya pengamatan yang seharusnya dilakukan, ini berarti bahwa data yang diambil sudah mencukupi. Kemudian dengan melihat harga Dn dari tabel diatas dan dibandingkan dengan harga Dn di tabel, maka dapat disimpulkan bahwa data-data tersebut diatas berdistribusi normal.

Selanjutnya untuk mendapatkan waktu baku tiap operasi job, dapat dilakukan dengan menjumlahkan waktu permesinan dengan waktu waktu

standard loading-unloading kemudian hasilnya dikalikan dengan banyaknya pesanan. Dalam hal ini job akan diproses menurut batchnya. Apabila terdapat waktu pergantian fixture, maka waktu ini ditambahkan ke dalam waktu total pengerjaan operasi job (dalam batch).

Selanjutnya data-data mengenai job dan mesin yang terlibat dalam penjadwalan job shop ini perlu dinotasikan dengan cara memberi nomor untuk tiap job dan mesin. Pemberian nomor job-job adalah sebagai berikut :

Nomor Job	Nama Job
1	Intake mazda 626
2	Intake Mazda 323
3	Exhaust Mazda 323
4	Exhaust Mazda MR 90
5	Crank Case TS-230
6	Crank Case TF-300
7	Counter Weight F2L-912
8	Counter Weight F3L-912
9	Counter Weight F4L-912
10	Counter Weight F6L-912
11	Cylinder Head FL-913
12	Cylinder Head FL-912
13	Bearing Cap F2L-912
14	Trash Bearing F2L-912

Sedangkan untuk pemberian nomor mesin-mesin yang digunakan adalah sebagai berikut :

Nomor Mesin	Nama Mesin
1	VMC. 45 [43]
2	VMC. 40 [52]

3	Radial Drill [35]
4	Deboring
5	Jet Washer [72]
6	BMC. 40 [51]
7	Radial Drill [34]
8	BMC. 40 [53]
9	VMC. 45 [41]
10	NC H. Boring [150]
11	BMC. 80 [13]
12	BMC. 80 [16]
13	BMC. 80 [17]
14	BMC. 80 [12]
15	BMC. 80 [14]
16	BMC. 80 [15]
17	BMC. 80 [18]
18	BMC. 80 [11]
19	Rotary Grinding [410]
20	Vertical Milling [420]
21	Radial Drilling [430]
22	Engine Lathe [070]
23	Work Bench [440]
24	NC Lathe [62]
25	VMC. 45 [42]

Setelah pemberian nomor job dan mesin, selanjutnya adalah mendapatkan waktu total masing-masing operasi tiap job. Waktu total masing-masing operasi tiap job dapat diperoleh dengan menjumlahkan waktu operasi dengan waktu pergantian fixture. Sedangkan waktu operasi diperoleh dengan mengalikan banyaknya pesanan job dengan hasil penjumlahan dari waktu permesinan dengan waktu standard loading-unloading.

Waktu total operasi untuk masing-masing operasi job ditunjukkan dibawah ini :

Keterangan notasi :

Wp : Waktu proses permesinan

Wlu : Waktu standard loading-unloading

Wo : Waktu operasi job

Wpf : Waktu pergantian fixture

Wtot : Waktu total operasi job

Nomor Job	Nomor Operasi	Nomor Mesin	Wp (menit)	Wlu (menit)	Wo (jam)	Wpf (jam)	Wtot (jam)
1	1	1	10	4,62	14,62	-	14,62
	2	2	20	2,28	22,28	-	22,28
	3	1	15	2,97	17,97	3,85	21,82
	4	1	15	2,75	17,75	-	17,75
	5	3	10	0,97	10,97	-	10,97
	6	3	6	0,72	6,72	-	6,72
	7	4	5	-	5	-	5
	8	5	10	3,46	0,89	-	0,89
2	9	6	10	2,23	16,31	-	16,31
	10	6	20	2,42	29,89	2,32	32,22
	11	1	20	2,28	29,71	3,43	33,14
	12	7	5	1,09	8,12	-	8,12
	13	7	5	0,83	7,77	-	8,12
	14	4	5	-	6,67	-	6,67
	15	5	10	3,44	1,19	-	1,19
3	16	8	15	2,29	23,05	-	23,05
	17	8	10	2,45	16,60	-	16,60
	18	7	5	0,56	7,41	1,17	8,58
	19	4	2	-	2,67	-	2,67
	20	5	10	3,32	1,18	-	1,18

Nomor Job	Nomor Operasi	Nomor Mesin	Wp (menit)	Wlu (menit)	Wo (jam)	Wpf (jam)	Wtot (jam)
4	21	9	20	3,42	35,13	-	35,13
	22	10	20	3,36	35,04	-	35,04
	23	7	1	0,58	4,74	-	4,74
	24	4	1,5	-	4,5	-	4,5
	25	5	10	3,67	2,73	-	2,73
5	26	11	31	5,41	30,34	-	30,34
	27	12	46	4,94	42,45	-	42,45
	28	13	43	5,18	40,15	-	40,15
	29	14	46	5,36	42,8	-	42,80
	30	5	32	11,98	2,73	-	5,24
6	31	15	32	9,20	27,47	-	27,47
	32	16	51	9,15	40,10	-	40,10
	33	17	52	9,49	40,99	-	40,99
	34	18	47	9,20	37,47	-	37,47
	35	5	31	11,52	4,72	-	4,72
7	36	19	32	5,76	2,62	-	2,62
	37	20	4,2	2,35	10,92	-	10,92
	38	21	8,2	2,44	17,73	-	17,73
	39	22	2,6	2,11	7,85	-	7,85
	40	23	2	-	3,33	-	3,33
8	41	19	35	5,72	3,39	-	3,39
	42	20	4,4	2,33	13,46	2,3	15,76
	43	21	8,2	1,92	20,24	1,14	21,38
	44	23	2	-	4	-	3,33
9	45	19	33,7	5,72	2,46	-	2,46
	46	20	4,7	2,37	10,61	2,3	15,76
	47	21	8,2	2,19	15,59	1,14	16,73
	48	23	2	-	3	-	3

Nomor Job	Nomor Operasi	Nomor Mesin	Wp (menit)	Wlu (menit)	Wo (jam)	Wpf (jam)	Wtot (jam)
10	49	19	31,6	5,6	2,20	-	2,20
	50	20	4,2	2,35	9,28	2,3	11,58
	51	21	8,2	2,34	14,93	1,14	16,07
	52	23	2	-	2,83	-	2,83
11	53	6	24	3,42	21,02	-	21,02
	54	24	2,28	0,58	2,19	-	2,19
	55	6	10,5	6,14	12,76	4,60	17,36
	56	5	20	6,42	0,81	-	0,81
12	57	6	24	3,46	57,21	-	57,21
	58	24	2,28	0,58	5,96	-	5,96
	59	6	10,5	5,94	34,25	4,60	38,85
	60	5	20	6,56	2,21	-	2,21
13	61	19	15	5,51	1,81	-	1,81
	62	25	59,76	6,07	46,45	-	46,45
	63	5	25	11,1	0,76	-	0,76
14	64	19	30	5,97	2,5	-	2,5
	65	25	25	5,69	17,05	3,45	20,5
	66	5	25	11,17	0,6	-	0,6

BAB VI

ANALISIS MODEL

Setelah dibentuk model dasar dan pengembangannya, maka pada bab ini akan dijelaskan tentang analisis model. Analisis dilakukan terhadap performansi yang ditunjukkan model dasar dalam menyelesaikan setiap kasus penjadwalan job shop dengan kriteria minimasi makespan. Performansi yang dianalisa dalam hal ini adalah nilai makespan yang dihasilkan oleh model dan waktu yang diperlukan untuk menghasilkan jadwal dengan nilai makespan tersebut. Kasus-kasus yang akan diselesaikan oleh metoda shifting bottleneck digenerate oleh penulis dengan setiap kasus memiliki variabel jumlah job, operasi dan mesin yang berbeda-beda. Juga akan dianalisis adanya routing job yang kompleks, waktu siap job dan mesin yang tidak harus sama dengan nol serta adanya job yang berprioritas. Kemudian juga dianalisis penjadwalan untuk data-data yang diambil dari PT. BBI

6.1. Analisis Terhadap Model Dasar

Model dasar yang telah dibentuk untuk menyelesaikan permasalahan penjadwalan job shop dengan kriteria minimasi makespan yang telah dijelaskan dalam bab terdahulu, akan dianalisis performansinya dalam menyelesaikan berbagai kasus penjadwalan dengan N job M mesin. Setiap kasus yang diselesaikan oleh model dasar dibedakan berdasarkan variabel input yang akan dimasukkan ke dalam model. Variabel-variabel tersebut meliputi jumlah mesin,

proses untuk setiap operasi. Bahasa pemrograman yang digunakan adalah Turbo

Pascal versi 7.0 yang dijalankan pada komputer yang memiliki CPU 386 SX-24

Tabel 6.1. Hasil Penjadwalan Dari Model Dasar Untuk Berbagai Kasus

Kasus	Mesin	Job	Operasi	Makespan	Wkt Proses (detik)	Lower Bound
1	4	5	3	13*	0.11	13*
2	5	5	5	117*	0.28	117*
3	5	6	5	109	0.39	96
4	5	10	5	238	0.6	237
5	5	12	5	585	0.83	492
6	6	5	6	72	0.44	63
7	10	5	8	87	1.04	79
8	10	6	8	211	1.26	175
9	10	8	8	233	1.65	175
10	10	8	10	899	2.19	722
11	10	10	10	588	2.91	398
12	15	10	15	883	7.52	674
13	20	5	20	1121	7.14	1008
14	20	10	20	1537	15.27	989
15	20	20	5	665	7.45	518
16	25	14	8	184.46	7.25	183.78

URN

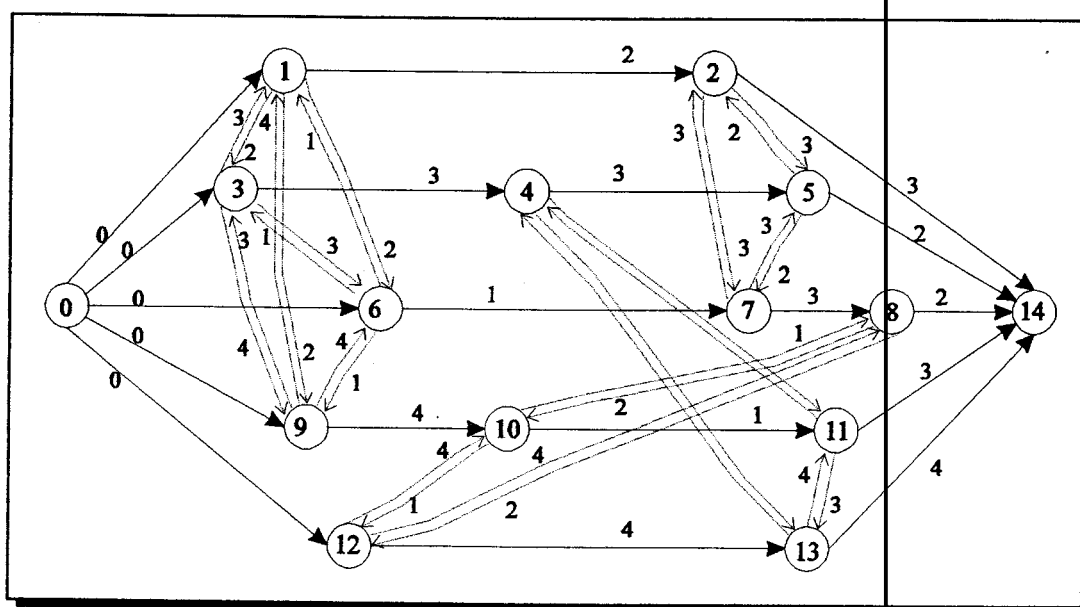
9832010016/28 Juni 2001

Dari tabel 6.1. dapat dilihat bahwa model dasar yang dibentuk untuk menyelesaikan permasalahan penjadwalan job shop dengan kriteria minimasi makespan diuji untuk menyelesaikan berbagai kasus penjadwalan, mulai dari 4 mesin, 5 job dan 3 operasi sampai 25 mesin, 14 job dan 20 operasi. Nilai lower bound yang muncul untuk setiap kasus diatas menunjukkan nilai maksimum makespan yang didapat dari penjadwalan satu mesin pada saat belum ada mesin yang dijadwalkan atau ketika $M_o = \phi$. Sementara itu jika ada nilai makespan akhir ketika semua mesin telah terjadwal yang besarnya sama dengan nilai lower

boundnya, maka nilai makespan tersebut merupakan nilai makespan yang optimal. Akan tetapi jika makespan yang dihasilkan lebih besar dari lower boundnya, maka makespan tersebut merupakan hasil dari jadwal yang dibentuk oleh algoritma Shifting Bottleneck yang mungkin saja merupakan solusi optimal. Dari 16 kasus yang diujikan terhadap model, ada dua kasus yang menghasilkan solusi optimal yaitu kasus nomor 1 dan nomor 2 yang dihasilkan oleh metoda shifting bottleneck.

Waktu proses atau komputasi yang diperlukan metoda shifting bottleneck untuk menghasilkan solusi tampak semakin besar dengan semakin besarnya masalah yang dihadapi. Jumlah job dan jumlah operasi menunjukkan banyaknya simpul dalam graph yang akan mempengaruhi berapa kali algoritma mencari lintasan terpanjang yang harus ditempuh, sedangkan banyaknya mesin membuat prosedur pencarian mesin bottleneck dan reoptimasi lokal akan terpengaruhi jumlah langkahnya dalam melakukan penjadwalan satu mesin. Faktor-faktor ini akan mempengaruhi waktu proses komputasi yang dibutuhkan oleh metoda shifting bottleneck dalam menghasilkan solusi.

Salah satu kasus yang telah diselesaikan oleh model dasar adalah kasus penjadwalan untuk 4 mesin 5 job dengan operasi maksimum tiap job 3 buah. Selanjutnya permasalahan tersebut dijelaskan dalam disjunctive graph dibawah ini



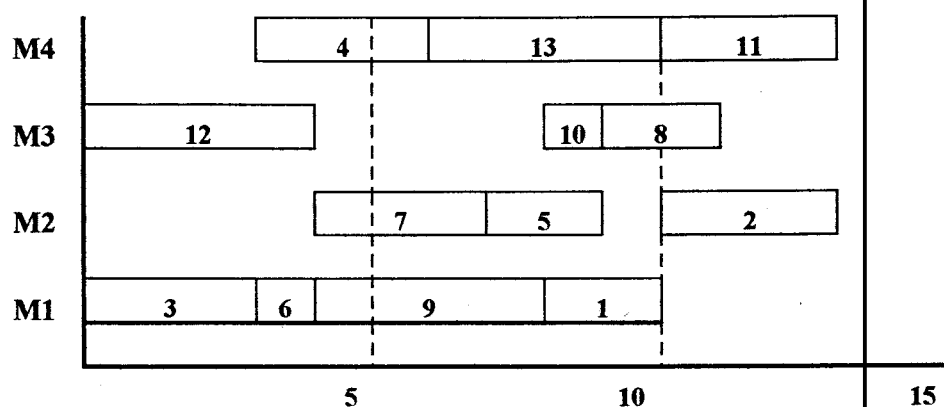
Gambar 6.1. Disjunctive Graph Model Dasar

Tabel 6.2. Input Penjadwalan Model Dasar

Nomor Job	Nomor Operasi	Waktu Proses	Nomor Mesin
1	1	2	1
	2	3	2
2	3	3	1
	4	3	4
	5	2	2
3	6	1	1
	7	3	2
	8	2	3
4	9	4	1
	10	1	3
	11	3	4
5	12	4	3
	13	4	4

Tabel 6.3. Hasil Penjadwalan Model Dasar

Nomor Mesin	Nomor Job	Nomor Operasi	Waktu Proses	Waktu Mulai	Waktu Selesai
1	2	3	3	0	3
1	3	6	1	3	4
1	4	9	4	4	8
1	1	1	2	8	10
2	3	7	3	4	7
2	2	5	2	7	9
2	1	2	3	10	13
3	5	12	4	0	4
3	4	10	1	8	9
3	3	8	2	9	11
4	2	4	3	3	6
4	5	13	4	6	10
4	4	11	3	10	13



Gambar 6.2. Gantt Chart Hasil Model Dasar

Dari gambar 6.2. tampak bahwa Gantt Chart yang diplot berdasarkan hasil penjadwalan yang diselesaikan oleh metoda Shifting Bottleneck sudah menunjukkan suatu jadwal yang layak dan urutan atau routing dari masing-masing job sudah memenuhi precedent constraint. Dari Gant Chart tersebut dapat dilihat

bahwa waktu yang diperlukan untuk menyelesaikan seluruh job atau makespannya sama dengan 13 yang ditandai dengan selesainya operasi 2 pada mesin 2 dan operasi 11 pada mesin 4.

6.2. Analisis Perbandingan Dengan Model Lain

Guna mengetahui performansi yang ditunjukkan oleh model dasar Shifting Bottleneck dalam menyelesaikan masalah penjadwalan job shop dengan kriteria minimasi makespan, maka disini model dasar akan dibandingkan metoda lain, yaitu Quant System versi 1.0. Performansi yang akan dibandingkan dalam hal ini adalah nilai makespan yang dihasilkan oleh kedua metoda tersebut dan kecepatan proses perhitungan yang diperlukan untuk menghasilkan jadwal.

Berikut ini merupakan hasil dari perbandingan kedua metoda tersebut dalam menyelesaikan permasalahan penjadwalan job shop.

Tabel 6.2. Perbandingan Shifting Bottleneck Dengan Priority Dispatching

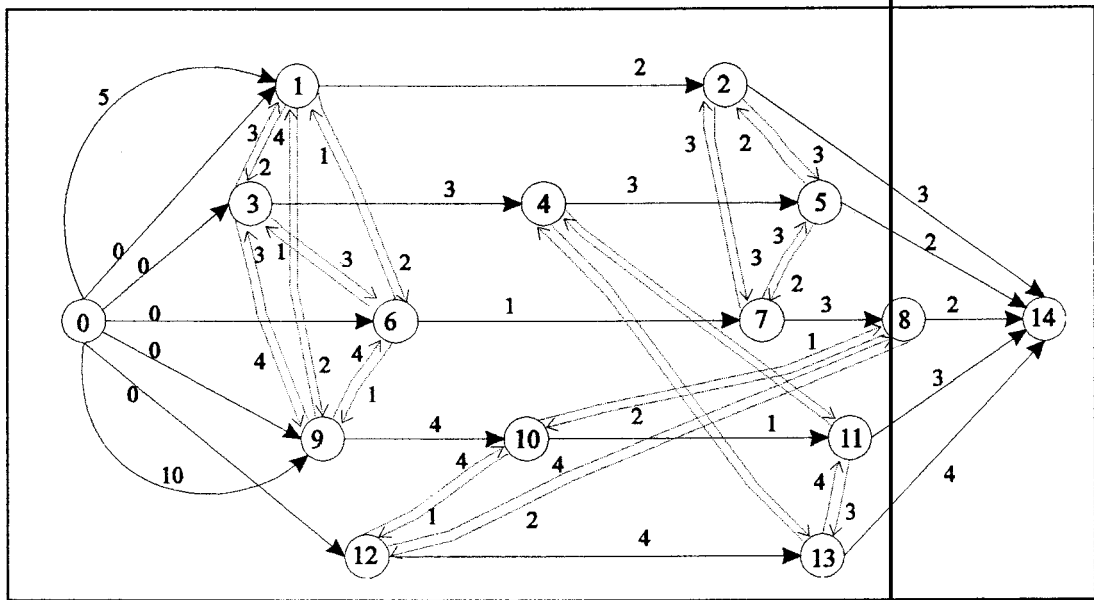
Kasus	M	J	O	Shifting Bottleneck			Priority Quant System	
				Makespan	LB	Waktu	Makespan	Wakt
1	4	5	3	13	13	0.11	14	1.25
2	5	5	5	117	117	0.28	117	1.88
3	5	10	5	238	237	0.6	244	4.10
4	6	5	6	72	63	0.44	72	0.81
5	10	5	8	87	79	1.04	87	1.19
6	10	8	10	899	722	2.19	923	4.16
7	10	5	10	112	95	1.26	115	1.5
8	15	10	15	883	674	7.52	984	1.59
9	20	20	5	665	518	7.47	677	4.25
10	25	14	8	184.46	183.78	7.25	184.78	11.15

Perbandingan terhadap model lain dalam hal ini teknik priority dispatching dilakukan untuk menunjukkan nilai solusi yang dihasilkan dan waktu komputasi yang diberikan oleh masing-masing metoda. Dari tabel tersebut tampak bahwa secara umum metoda shifting bottleneck lebih baik jika dibandingkan dengan teknik priority dispatching. Dari 10 kasus yang masing-masing diselesaikan oleh kedua metoda, tampak bahwa 6 kasus dapat diselesaikan oleh metoda shifting bottleneck dengan nilai makespan yang lebih baik dan waktu komputasinya juga relatif lebih cepat, kecuali untuk kasus 8 dan kasus 9 dengan jumlah job sama dengan 10 dan 20, jumlah mesin 15 dan 20 serta jumlah operasi 15 dan 5. Kelemahan yang muncul disini adalah waktu komputasi yang relatif lebih lama jika dibandingkan dengan teknik priority dispatching untuk permasalahan yang semakin besar, akan tetapi hal ini dapat diimbangi dengan nilai solusi dengan makespan yang lebih baik dan waktu komputasi yang lebih cepat untuk permasalahan yang kecil.

6.3. Analisis Model Untuk Job Yang Tidak Siap Pada $t = 0$

Model dasar dikembangkan untuk menyelesaikan masalah penjadwalan job shop yang memperhatikan waktu siap dari job-job yang akan dijadwalkan tidak harus sama dengan nol pada awal perioda penjadwalan. Adanya waktu siap dari job-job yang akan dijadwalkan tidak sama dengan nol dapat saja terjadi. Hal ini karena mungkin saja material dari job-job tersebut belum datang pada awal perioda penjadwalan. Model yang dikembangkan dapat menyelesaikan masalah diatas dengan batasan atau asumsi bahwa waktu siap dari job-job yang akan

dijadwalkan diketahui terlebih dahulu secara deterministik. Analisa terhadap model dilakukan dengan melakukan pengujian terhadap beberapa problem dan salah satu hasil penjadwalan yang telah diselesaikan oleh model ditampilkan dalam Gantt Chart.



Gambar 6.3. Disjunctive Graph Job Tak Siap

Salah satu problem yang telah diselesaikan adalah masalah penjadwalan dengan 4 mesin 5 job dengan banyaknya operasi tiap job maksimum 3 buah serta job 1 dan job 4 masing-masing waktu siapnya adalah 5 dan 10. Pengembangan model untuk menyelesaikan masalah ini dilakukan dengan cara menambahkan dua busur yang menghubungkan simpul nol ke operasi pertama dari job 1 dan job 4. Selanjutnya permasalahan ini dapat dilihat pada gambar 6.3

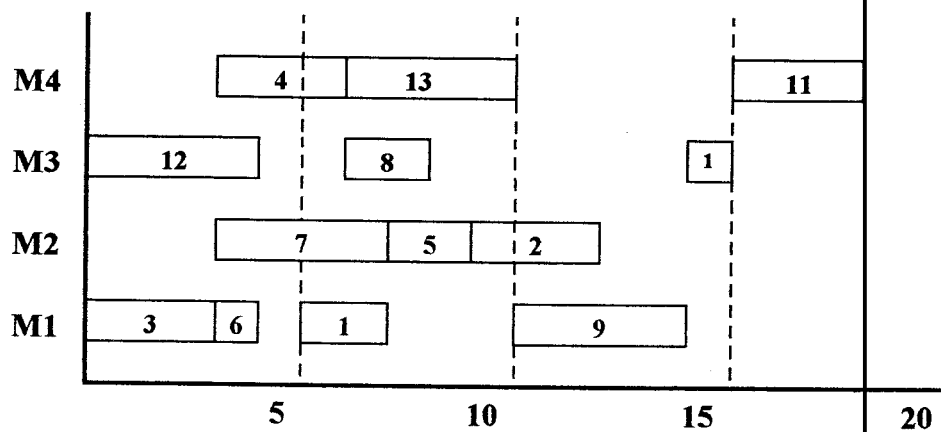
Tabel 6.5. Input Penjadwalan Kasus

Job Tak Siap Pada $t=0$

Nomor Job	Nomor Operasi	Waktu Proses	Waktu Siap	Nomor Mesin
1	1	2	5	1
	2	3	0	2
2	3	3	0	1
	4	3	0	4
	5	2	0	2
3	6	1	0	1
	7	3	0	2
	8	2	0	3
4	9	4	10	1
	10	1	0	3
	11	3	0	4
5	12	4	0	3
	13	4	0	4

Tabel 6.3. Hasil Penjadwalan Job Tak Siap Pada $t=0$

Nomor Mesin	Nomor Job	Nomor Operasi	Waktu Proses	Waktu Mulai	Waktu Selesai
1	2	3	3	0	3
1	3	6	1	3	4
1	1	1	2	5	7
1	4	9	4	10	14
2	3	7	3	4	7
2	2	5	2	7	9
2	1	2	3	9	12
3	5	12	4	0	4
3	3	8	2	7	9
3	4	10	1	14	15
4	2	4	3	3	6
4	5	13	4	6	10
4	4	11	3	15	18



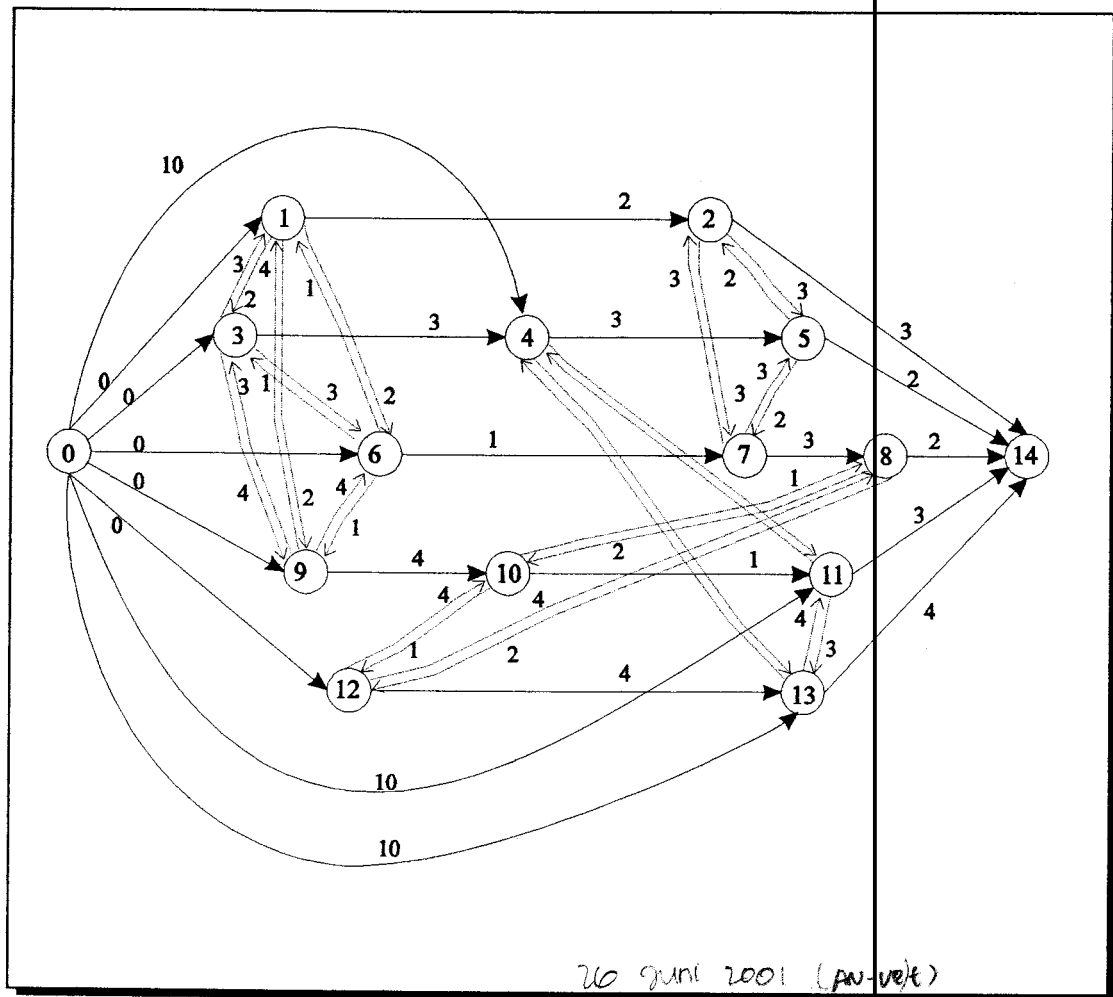
Gambar 6.4. Gantt Chart Hasil Jadwal Job Tak Siap

Dari Gantt Chart diatas dapat dilihat bahwa operasi 1 yang mewakili operasi pertama dari job 1 diproses di mesin 1 pada $t = 5$, padahal pada saat $t = 4$ yaitu setelah operasi nomor 6 selesai, mesin 1 dalam keadaan idle yang sebenarnya operasi nomor 1 dapat diproses pada saat itu. Berhubung operasi 1 siap pada $t = 5$, maka operasi 1 baru diproses di mesin 1 pada $t = 5$. Demikian juga untuk operasi nomor 9 yang mewakili operasi pertama dari job 4 yang siap pada $t = 10$, akan diproses pada $t = 10$ di mesin 1.

Solusi yang dihasilkan untuk kasus ini menunjukkan bahwa model yang telah dikembangkan dapat menyelesaikan masalah penjadwalan dengan waktu siap yang tidak sama dengan nol. Penambahan busur yang menunjukkan kesiapan suatu job akan mempengaruhi lamanya perhitungan lintasan terpanjang untuk pencarian lintasan kritisnya yang merupakan makespan dari permasalahan tersebut. Tetapi kelemahan ini dapat diatasi dengan menerapkan algoritma yang tepat, misalnya algoritma pencarian rute terpendek yang telah dimodifikasi untuk pencarian rute atau lintasan terpanjang.

6.4. Analisis Model Untuk Mesin Yang Tidak Siap Pada $t = 0$

Pengembangan model dasar yang digunakan untuk menyelesaikan masalah penjadwalan job shop dengan memperhatikan adanya mesin yang tidak siap pada awal perioda penjadwalan, pada dasarnya sama dengan model yang dikembangkan untuk penyelesaian penjadwalan dengan job yang tidak siap pada $t = 0$. Perbedaannya hanya terletak pada penambahan busur yang diberikan pada model. Untuk menyatakan kesiapan suatu mesin dilakukan penambahan busur yang menghubungkan simpul nol atau awal ke masing-masing operasi yang dikerjakan oleh mesin yang tidak siap tersebut. Selanjutnya untuk memperjelas permasalahan ini kita dapat melihat pada gambar disjunctive graph 6.5..



Gambar 6.5. Disjunctive Graph Mesin Tak Siap

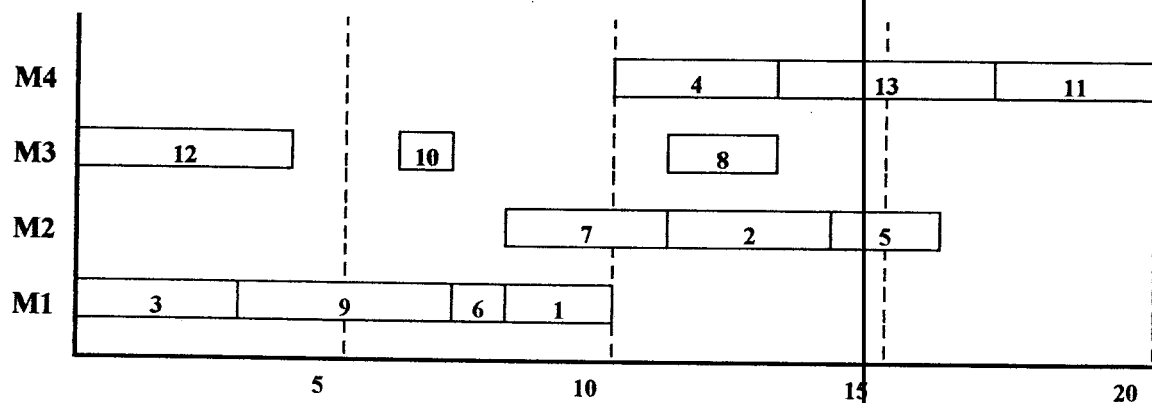
Salah satu permasalahan yang telah diselesaikan oleh model ini adalah masalah penjadwalan 4 mesin 5 job dengan operasi maksimum 3 buah per job. Mesin ke 4 pada kasus ini siap pada $t = 10$, sedangkan mesin yang lain siap pada $t = 0$. Sehingga kita perlu melakukan penambahan busur yang menghubungkan simpul nol ke simpul atau operasi nomor 4, 11 dan 13 dimana operasi-operasi tersebut dikerjakan oleh mesin 4.

Tabel 6.4. Input Penjadwalan Model Mesin Tak Siap

Nomor Job	Nomor Operasi	Waktu Proses	Nomor Mesin	Waktu Siap Mesin
1	1	2	1	0
	2	3	2	0
2	3	3	1	0
	4	3	4	10
	5	2	2	0
3	6	1	1	0
	7	3	2	0
	8	2	3	0
4	9	4	1	0
	10	2	3	0
	11	3	4	10
5	12	4	3	0
	13	4	4	10

Tabel 6.8. Hasil Penjadwalan Mesin Tak Siap

Nomor Mesin	Nomor Job	Nomor Operas	Waktu Proses	Waktu Mulai	Waktu Selesai
1	2	3	3	0	3
1	4	9	4	3	7
1	3	6	1	7	8
1	1	1	2	8	10
2	3	7	3	8	11
2	1	2	3	11	14
2	2	5	2	14	16
3	5	12	4	0	4
3	4	10	1	7	8
3	3	8	2	11	13
4	2	4	3	10	13
4	5	13	4	13	17
4	4	11	3	17	20



Gambar 6.6. Gantt Chart Hasil Jadwal Mesin Tak Siap

Dari gantt chart yang diplot berdasarkan hasil jadwal yang dikeluarkan oleh model, kita dapat melihat bahwa operasi nomor 4 mulai diproses di mesin 4 pada $t = 10$ dimana sebenarnya pada saat $t = 3$, operasi 4 dapat segera diproses. Tetapi karena mesin 4 siap pada $t = 10$, maka operasi-operasi yang dikerjakan oleh mesin tersebut (termasuk operasi 11 dan operasi 13) akan mulai dikerjakan pada waktu siap dari mesinnya, meskipun sebenarnya waktu mulai pemrosesan dari operasi-operasi dapat saja dimulai segera setelah operasi pendahulunya selesai.

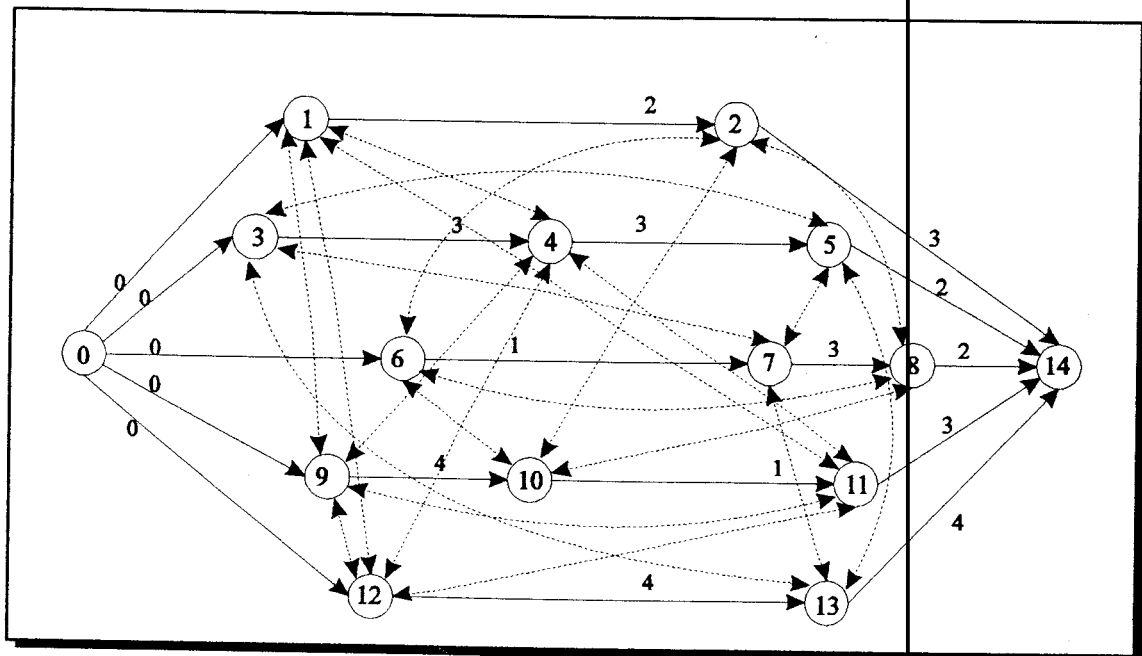
Solusi yang diberikan ini menunjukkan bahwa model telah dapat dipakai untuk menyelesaikan masalah penjadwalan dengan waktu siap mesin yang tidak sama dengan nol

6.5. Analisis Model Untuk Routing Job Yang Kompleks

Terkadang kita menemui satu atau beberapa job yang operasi-operasinya diproses pada mesin yang sama lebih dari satu kali dan keadaan ini akan

membentuk routing yang kompleks. Model dasar yang dikembangkan dapat menyelesaikan permasalahan ini, yaitu dengan cara melakukan mekanisme kontrol terhadap job-job yang diproses pada mesin yang sama lebih dari satu kali agar operasi-operasi dari job tersebut tidak menyalahi aturan precedent constraint.

Salah satu permasalahan yang telah diselesaikan oleh model ini adalah penjadwalan 3 mesin 5 job dimana job 2 diproses pada mesin 3 sebanyak dua kali, job 3 diproses pada mesin 2 sebanyak dua kali dan job 4 diproses pada mesin 1 juga sebanyak dua kali. Selanjutnya agar lebih jelas dapat dilihat pada gambar disjunctive graph untuk routing kompleks ini



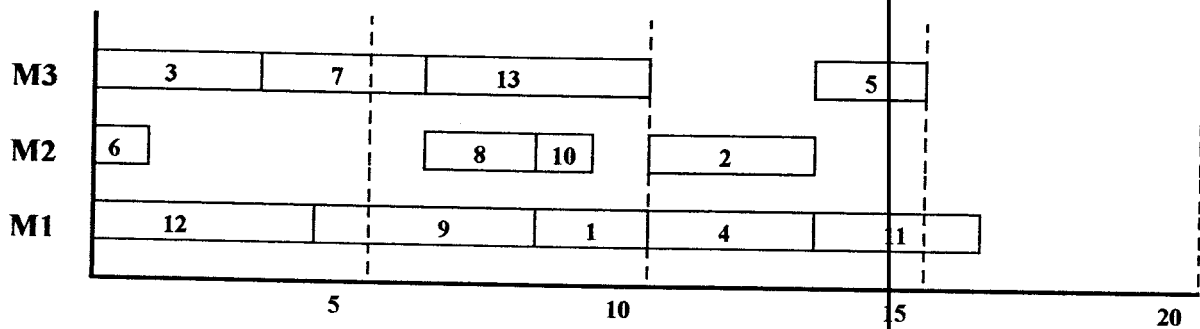
Gambar 6.7. Disjunctive Graph Routing Komplek

Tabel 6.9. Input Penjadwalan Routing Komplek

Nomor Job	Nomor Operasi	Waktu Proses	Nomor Mesin
1	1	2	1
	2	3	2
2	3	3	3
	4	3	1
	5	2	3
3	6	1	2
	7	3	3
	8	2	2
4	9	4	1
	10	1	2
	11	3	1
5	12	4	1
	13	4	3

Tabel 6.10. Hasil Penjadwalan Model Routing Komplek

Nomor Mesin	Nomor Job	Nomor Operasi	Waktu Proses	Waktu Mulai	Waktu Selesai
1	5	12	4	0	4
1	4	9	4	4	8
1	2	4	3	10	13
1	4	11	3	13	16
2	3	6	1	0	1
2	3	8	2	6	8
2	4	10	1	8	9
2	1	2	3	10	13
3	2	3	3	0	3
3	3	7	3	3	6
3	5	13	4	6	10
3	2	5	2	13	15



Gambar 6.8. Gantt Chart Hasil Jadwal Routing Komplek

Dari Gantt Chart yang diplot berdasarkan hasil penjadwalan yang dikeluarkan oleh model, maka dapat dijelaskan bahwa operasi-operasi yang membentuk suatu job yang diproses pada suatu mesin telah memenuhi hubungan precedent constraint. Misalnya untuk job 2 yang mempunyai dua operasi yaitu operasi nomor 3 dan operasi nomor 5 yang keduanya diproses pada mesin 3, telah memenuhi hubungan precedent constraint dan untuk job-job lain yang diproses pada mesin yang sama, operasi-operasinya juga memenuhi hubungan tersebut.

Solusi yang diberikan untuk permasalahan ini menunjukkan bahwa model dasar yang telah dikembangkan mampu menyelesaikan penjadwalan job shop dengan routing komplek.

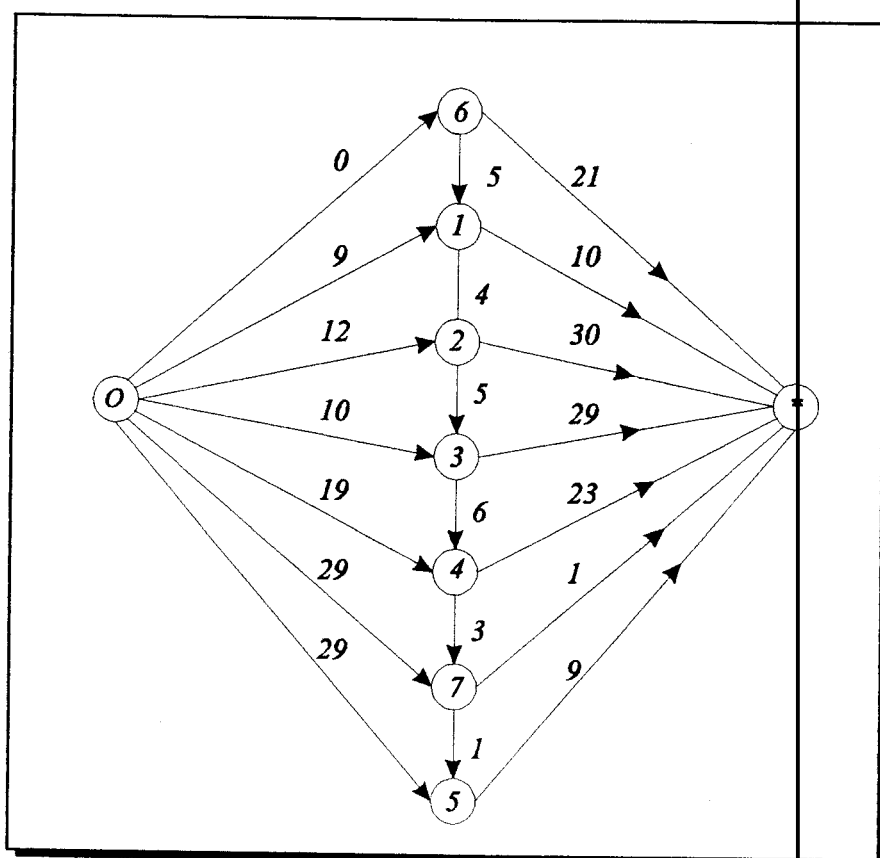
6.6. Analisis Model Untuk Job Berprioritas

Untuk menunjukkan kemampuan model dalam menyelesaikan penjadwalan job shop dengan memperhatikan job-job yang berprioritas, maka disini akan diberikan satu contoh persoalan penjadwalan satu mesin dengan data-data sebagai berikut:

Operasi	1	2	3	4	5	6	7
r_i	9	12	10	19	29	0	29
d_i	4	5	6	3	2	5	1
q_i	6	25	23	20	7	16	0
Prioritas	2	1	2	2	2	1	1

Dari data tersebut, maka kita dapat menerapkan algoritma Schrage yang telah dimodifikasi untuk penjadwalan satu mesin dengan memperhatikan prioritas dari job yang ada. Modifikasi dari algoritma Schrage ini telah dijelaskan pada Bab IV.

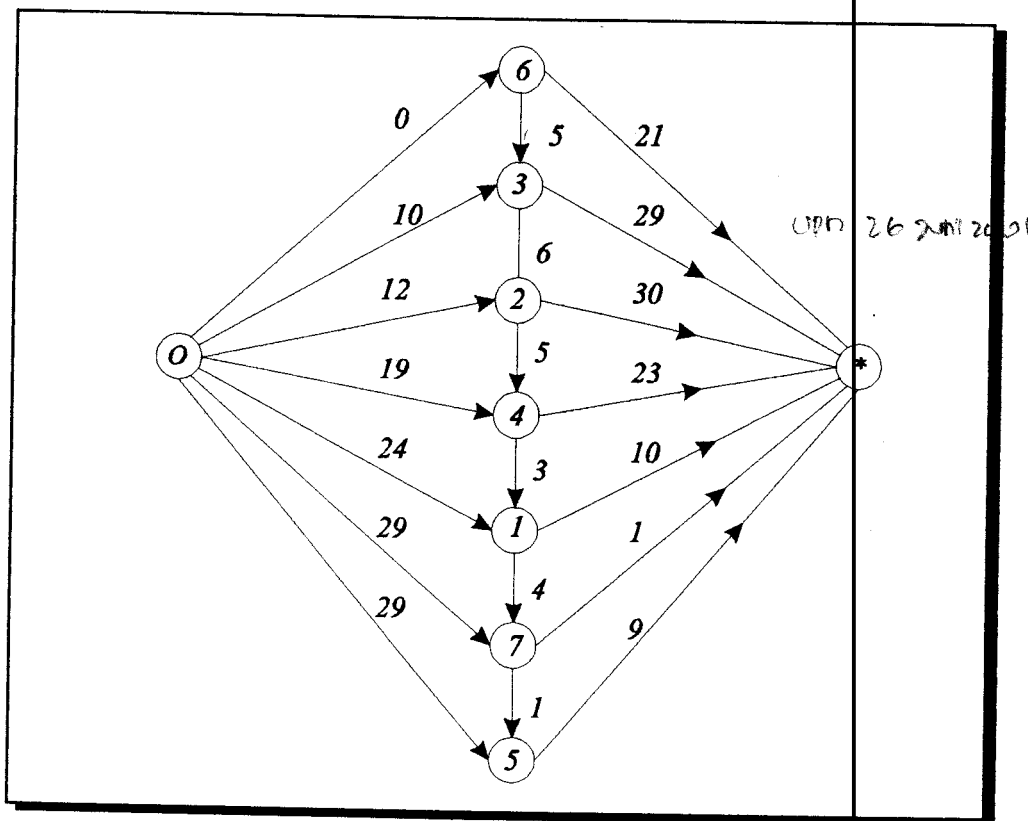
Selanjutnya dengan memakai modifikasi algoritma Schrage ini, kita dapatkan graph seperti pada gambar dibawah.



Gambar 6.9. Graph Job Berprioritas

Lintasan kritis yang terbentuk dari graph tersebut adalah 0-1-2-3-4-* dan nilainya sebesar 47. Dari simpul-simpul yang membentuk lintasan kritis ini ternyata simpul atau operasi terakhir $p = \text{simpul 4}$ dengan $q_p = 20$, maka simpul atau operasi kritis c adalah operasi 1 dan himpunan operasi J adalah operasi 2, 3 dan 4. Selanjutnya dengan teknik branch and bound, akan diperiksa apakah operasi kritis c dapat dikerjakan setelah operasi-operasi himpunan J . Prioritas operasi 1 ternyata lebih kecil atau sama dengan prioritas operasi-operasi anggota himpunan J , maka operasi 1 dapat dikerjakan setelah operasi-operasi anggota himpunan J .

Operasi 1 dikerjakan setelah operasi-operasi anggota himpunan J akan menghasilkan graph seperti pada gambar berikut.



Gambar 6.10. Graph Solusi Job Berprioritas

Dari graph yang terbentuk akibat operasi 1 dikerjakan setelah operasi-operasi anggota himpunan J , akan terbentuk lintasan kritis baru yang melalui simpul 0-3-2-* yang nilai lintasan kritis atau makespannya dalam hal ini sama dengan 46. Solusi ini ternyata menghasilkan makespan yang lebih baik dari nilai makespan sebelumnya dan kita temukan dari lintasan kritis baru ini, operasi kritis $c = 3$ dan operasi anggota himpunan J adalah operasi 2. Selanjutnya dengan teknik branch and bound, kita periksa apakah operasi 3 dapat dikerjakan setelah operasi 2. Ternyata memang bisa, akan tetapi nilai makespan yang dihasilkan tidak lebih baik dari nilai makespan sebelumnya, maka urutan atau jadwal yang optimal untuk kasus ini adalah 6-3-2-4-1-7-5 dengan makespan 46.

Solusi yang diberikan untuk kasus ini menunjukkan bahwa model dapat menyelesaikan permasalahan penjadwalan job-job berprioritas.

6.7. Analisis Model Untuk Penjadwalan Di PT. BBI

Model yang dibentuk dan yang telah dikembangkan, dicoba diterapkan untuk menyelesaikan permasalahan penjadwalan job shop yang nyata dengan mengambil permasalahan penjadwalan di PT.BBI Surabaya. Data-data yang mengenai jumlah job, jumlah mesin, routing dari tiap job serta waktu prosesnya telah dijelaskan pada Bab IV dimuka.

Selanjutnya data-data tersebut dimasukkan ke model dan diselesaikan oleh model untuk menghasilkan suatu jadwal produksi. Solusi yang diberikan oleh model dala hal ini dapat dilihat pada tabel 6.11.

Tabel 6.11. Hasil Penjadwalan Di PT. BBI Surabaya (URN-APP - 98-54)

Nomor Job	Operasi Ke	Nomor Operasi	Waktu Proses	Waktu Mulai	Waktu Selesai	Nomor Mesin
1	1	1	14.62	0	14.62	1
	2	2	22.28	14.62	36.9	2
	3	3	21.28	36.9	58.72	1
	4	4	17.75	58.72	76.47	1
	5	5	10.97	76.47	87.44	3
	6	6	6.72	87.44	94.16	3
	7	7	5	94.16	99.16	4
	8	8	0.89	99.16	100.05	5
2	1	9	16.31	0	16.31	6
	2	10	32.22	94.54	126.76	6
	3	11	33.14	126.76	159.9	1
	4	12	8.12	159.9	168.02	7
	5	13	7.77	168.02	175.79	7
	6	14	6.67	175.79	182.46	4
	7	15	1.19	182.46	183.65	5
3	1	16	23.05	0	23.05	8
	2	17	16.6	23.05	39.65	8
	3	18	8.58	39.65	48.23	7
	4	19	2.67	48.23	50.9	4
	5	20	1.18	50.9	52.08	5
4	1	21	35.13	0	35.13	9
	2	22	35.04	35.13	70.17	10
	3	23	4.74	70.17	74.91	7
	4	24	4.5	74.91	79.41	4
	5	25	2.73	79.41	82.14	5
5	1	26	30.34	0	30.34	11
	2	27	42.45	30.34	72.79	12
	3	28	40.15	72.79	112.94	13
	4	29	42.8	112.94	155.74	14
	5	30	5.24	155.74	160.98	5
6	1	31	27.47	0	27.47	15
	2	32	40.1	27.47	67.57	16
	3	33	40.99	67.57	108.56	17
	4	34	37.47	108.56	146.03	18
	5	35	4.72	146.03	150.75	5
7	1	36	2.62	1.81	4.43	19

	2	37	10.92	10.32	21.24	20
	3	38	17.73	21.24	38.97	21
	4	39	7.85	38.97	46.82	22
	5	40	3.33	46.82	50.15	23
8	1	41	3.39	4.43	7.82	19
	2	42	15.76	21.24	37	20
	3	43	21.38	38.97	60.35	21
	4	44	3.33	60.35	63.68	23
9	1	45	2.46	7.82	10.28	19
	2	46	12.91	37	49.91	20
	3	47	16.73	60.35	77.08	21
	4	48	3	77.08	80.08	23
10	1	49	2.2	12.78	14.98	19
	2	50	11.58	49.91	61.49	20
	3	51	16.07	77.08	93.15	21
	4	52	2.83	93.15	95.98	23
11	1	53	21.02	73.52	94.54	6
	2	54	2.19	94.54	96.73	24
	3	55	17.36	165.61	182.97	6
	4	56	0.81	183.65	184.46	5
12	1	57	57.21	16.31	73.52	6
	2	58	5.96	73.52	79.48	24
	3	59	38.85	126.76	165.61	6
	4	60	2.21	165.61	167.82	5
13	1	61	1.81	0	1.81	19
	2	62	46.45	1.81	48.26	25
	3	63	0.76	48.26	49.02	5
14	1	64	2.5	10.28	12.78	19
	2	65	20.5	48.26	68.76	25
	3	66	0.6	68.76	69.36	5

Dari hasil penjadwalan yang ditunjukkan pada tabel 6.11., maka dapat diketahui bahwa total waktu yang diperlukan untuk menyelesaikan seluruh job sama dengan 184.46 jam dimana pada saat itu job 11 operasi ke 4 yang dikerjakan pada mesin 5 tepat selesai.

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Setelah melakukan analisis terhadap model yang diuji terhadap berbagai kasus permasalahan penjadwalan job shop, maka dalam penelitian ini penulis menyimpulkan bahwa :

1. Metoda Shifting Bottleneck dapat digunakan sebagai alternatif dalam pembuatan jadwal produksi job shop dengan kriteria minimasi makespan.
2. Metoda Shifting Bottleneck mempunyai solusi jadwal secara umum lebih baik jika dibandingkan dengan teknik priority dispatching terdapat dalam paket program Quant System dengan kriteria minimasi makespan.
3. Waktu komputasi yang diperlukan oleh metoda Shifting Bottleneck relatif lebih cepat dalam menghasilkan solusi. Waktu ini cenderung meningkat seiring dengan semakin besarnya masalah yang dihadapi, yaitu semakin banyak jumlah mesin dan operasi yang harus dijadwalkan.
4. Model dasar yang telah dikembangkan mampu untuk menyelesaikan masalah penjadwalan job shop dengan kasus-kasus seperti job dan mesin yang siap tidak sama dengan nol dan adanya job-job yang berprioritas serta routing yang kompleks dari suatu job yang terlibat dalam penjadwalan.

7.2. Saran

Saran-saran yang dapat diberikan oleh penulis adalah sebagai berikut :

1. Kriteria performansi dari jadwal yang dibentuk oleh metoda Shifting Bottleneck diharapkan dapat diperluas untuk penulisan lebih lanjut dengan kriteria performansi yang lain seperti mean flow time, number of tardy, mean tardiness dan lain sebagainya.
2. Pada keadaan sebenarnya, pola kedatangan dari job dan kesiapan mesin berlangsung secara dinamis dan keadaan ini tidak diperhatikan oleh penulis. Oleh karena itu perlu pengembangan model lebih lanjut untuk menyelesaikan permasalahan penjadwalan tersebut.
3. Metoda Shifting Bottleneck yang dikembangkan penulis diterapkan untuk menyelesaikan masalah penjadwalan job shop, oleh karena itu untuk karya penulisan lebih lanjut perlu dibentuk metoda shifting bottleneck baru yang diterapkan pada sistem produksi flow shop dan untuk N job M mesin yang dipasang secara paralel.

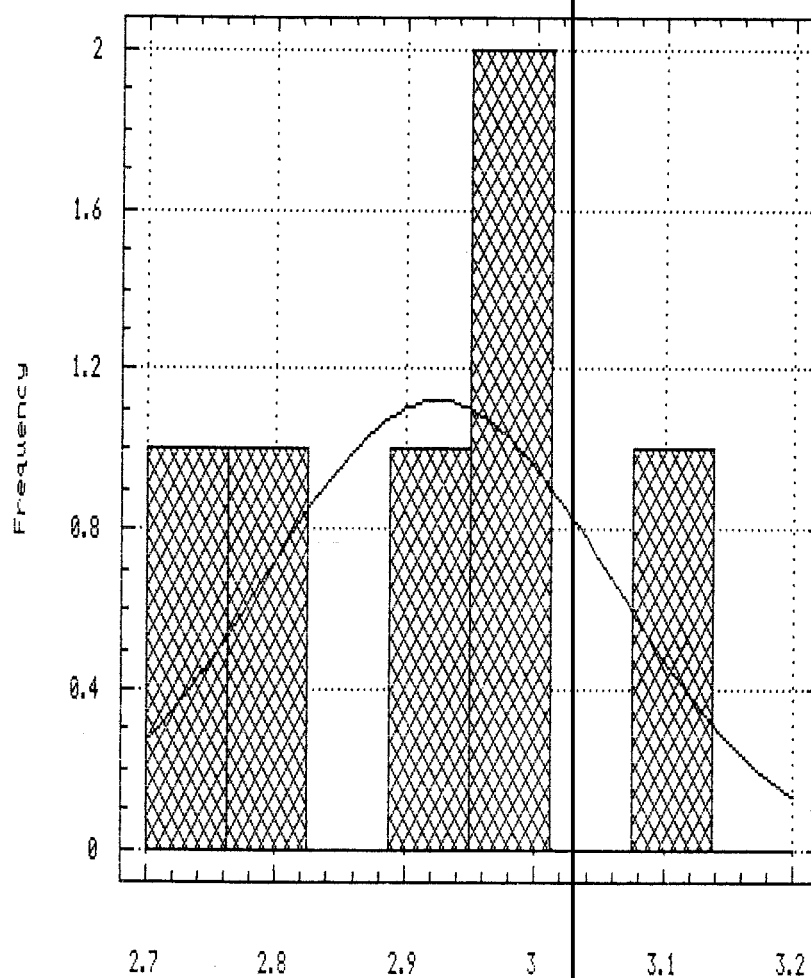
DAFTAR PUSTAKA

- Adams, Joseph. Egon Balas and Daniel Zawack, *The Shifting Bottleneck Procedure For Job Shop Scheduling*, Management Science Vol. 34, No. 3, March 1988
- Charlier, Jacques, *The One Machine Sequencing Problem*, European Journal of Operational Research 11 (1982) 42-47
- Barnes, Ralph M., *Motion and Time Study : Design and Measurement of Work*, John Wiley and Sons, New York, 1968
- Wignjosoebroto, Sritomo., *Teknik Tata Cara dan Pengukuran Kerja*, Institut Teknologi Sepuluh Nopember (ITS) Surabaya, 1989
- Walpole, Ronald E., and Raymond H Myers, *Ilmu Peluang Dan Statistika Untuk Insinyur Dan Ilmuwan*, Penerbit ITB, Bandung, 1986
- Daniel, Wayne W, *Statistika Non Parametrik Terapan*, PT. Gramedia, Jakarta, 1984
- Baker, Kenneth R., *Introduction to Sequencing and Scheduling*, John Wiley and Sons, New York, 1976
- Morton, Thomas E, and David W Pentico, *Heuristic Scheduling Systems*, John Wiley and Sons, New York, 1993
- Conway, Richard W, William L, Maxwell and Louis W. Millen, *Theory of Scheduling*, Addison-Wesley Publishing Company Ontario, 1967
- Bedworth, David D, and James E, Bailey, *Integrated Production Control System : Management, Analysis, Design*, John Wiley and Sons, New York 1987

- Elsayed, Elsayed A, and Thomas O Boucher, *Analysis and Control of Production System*, Prentice Hall, Inc, New Jersey, 1985
- Taha, Hamdy A, *Operation Research, An Introduction*, third edition, Macmillan Publishing Co, Inc, New York, 1982
- Law Averill M and N. David Kelton, *Simulation Modelling and Analysis*, McGraw-Hill, New York, Inc, 1982
- Jogiyanto H. M, *Teori dan Aplikasi Program Komputer Turbo Pascal Versi 5.0*, Jilid 1 dan 2, Andi Offset, Yogyakarta, 1989
- Wimanda, Rizki, *Perancangan Perangkat Lunak Untuk Penjadwalan Job Shop*, Tugas Akhir Jurusan Teknik Industri ITS, 1994.

Frequency Histogram

Proses 2 Exhaust Mazda MR90

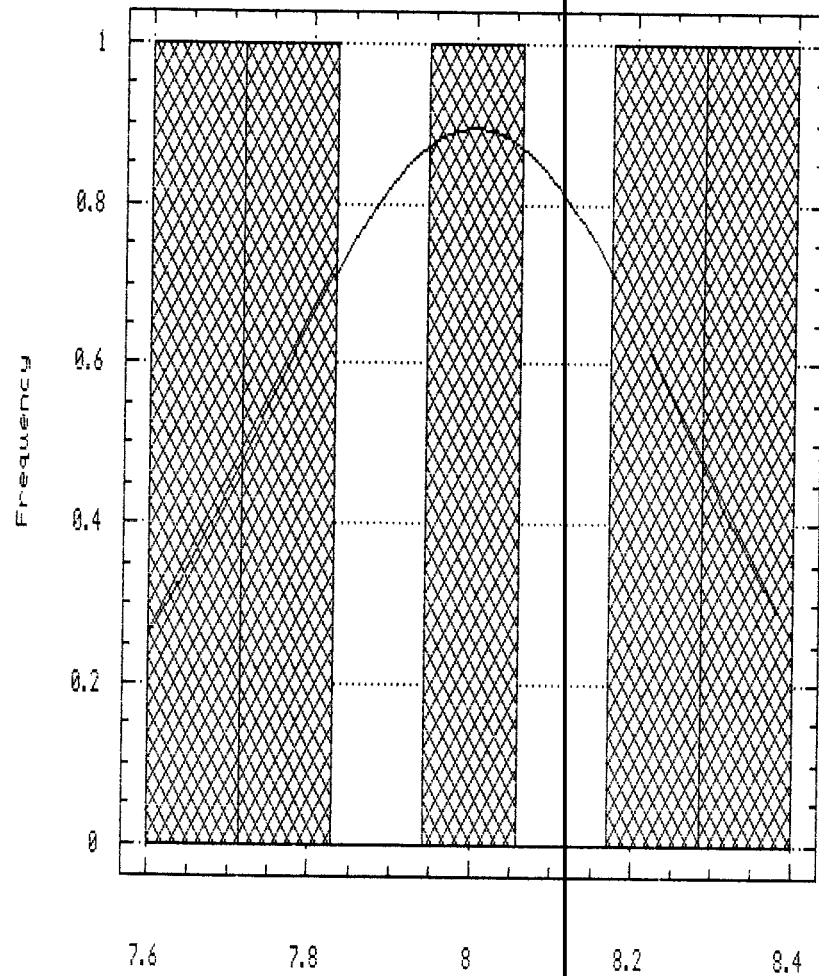


Waktu Loading-Unloading (menit)

Estimated KOLMOGOROV statistic DPLUS = 0.159272
 Estimated KOLMOGOROV statistic DMINUS = 0.213264
 Estimated overall statistic DN = 0.213264
 Approximate significance level = 0.947797

Frequency Histogram

Proses 1 Crank Case T7300

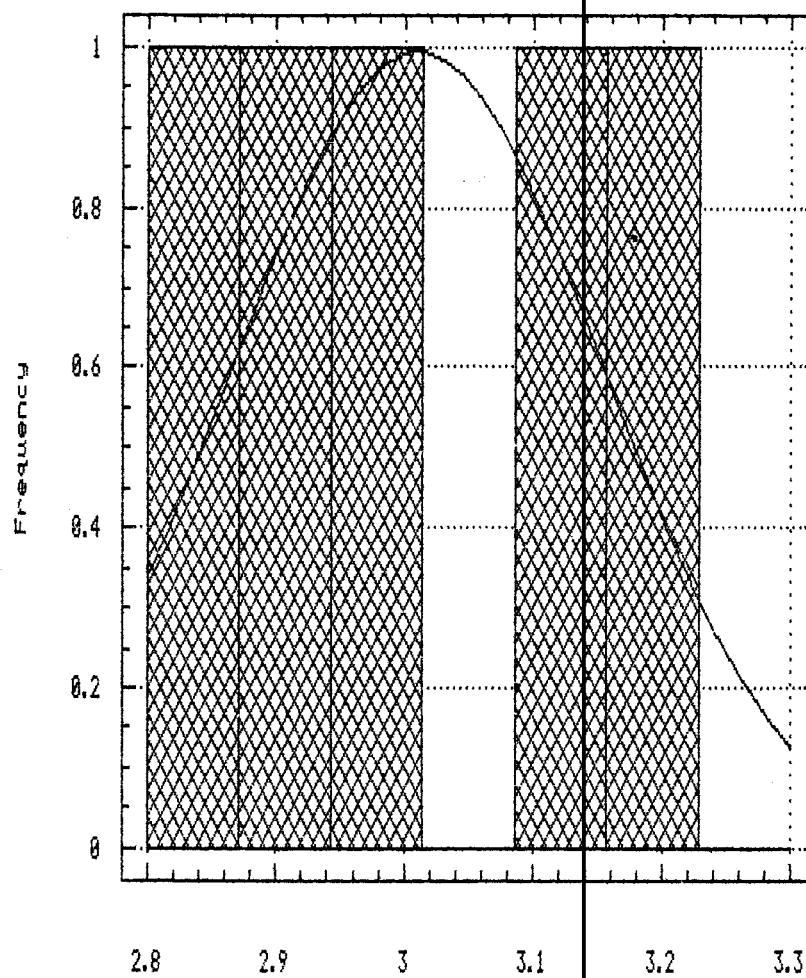


Waktu Loading-Unloading (menit)

Estimated KOLMOGOROV statistic DPLUS = 0.183617
 Estimated KOLMOGOROV statistic DMINUS = 0.183617
 Estimated overall statistic DN = 0.183617
 Approximate significance level = 0.995951

Frequency Histogram

Proses 1 Cylinder Head FL913

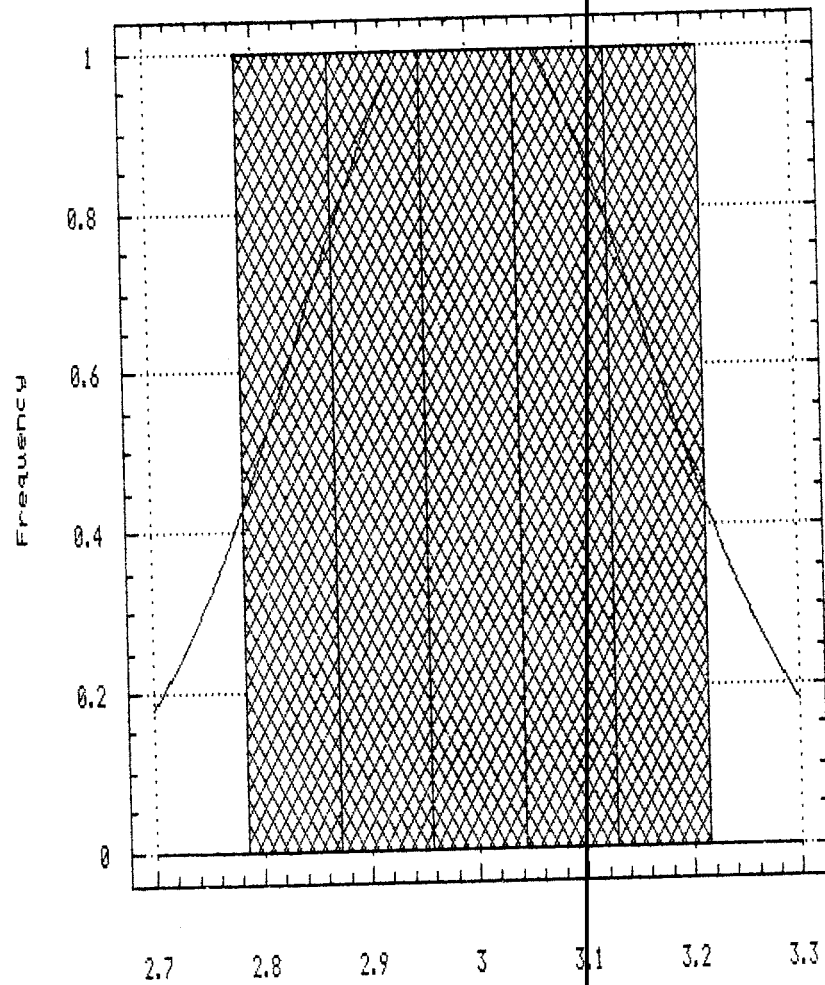


Waktu Loading-Unloading (menit)

Estimated KOLMOGOROV statistic DPLUS = 0.178839
 Estimated KOLMOGOROV statistic DMINUS = 0.135192
 Estimated overall statistic DN = 0.178839
 Approximate significance level = 0.997203

Frequency Histogram

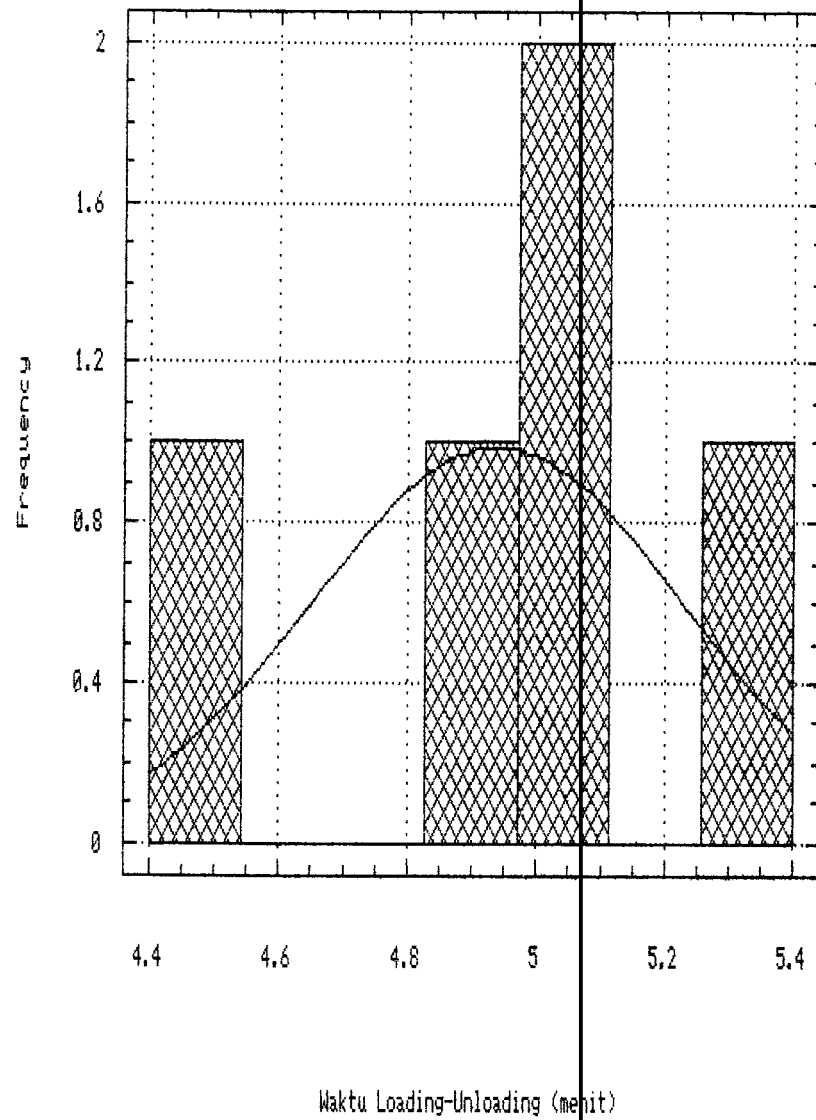
Ganti ke 1 Fixture



Waktu Pergantian Fixture (menit)

Estimated KOLMOGOROV statistic DPLUS = 0.136457
 Estimated KOLMOGOROV statistic DMINUS = 0.136457
 Estimated overall statistic DN = 0.136457
 Approximate significance level = 0.999986

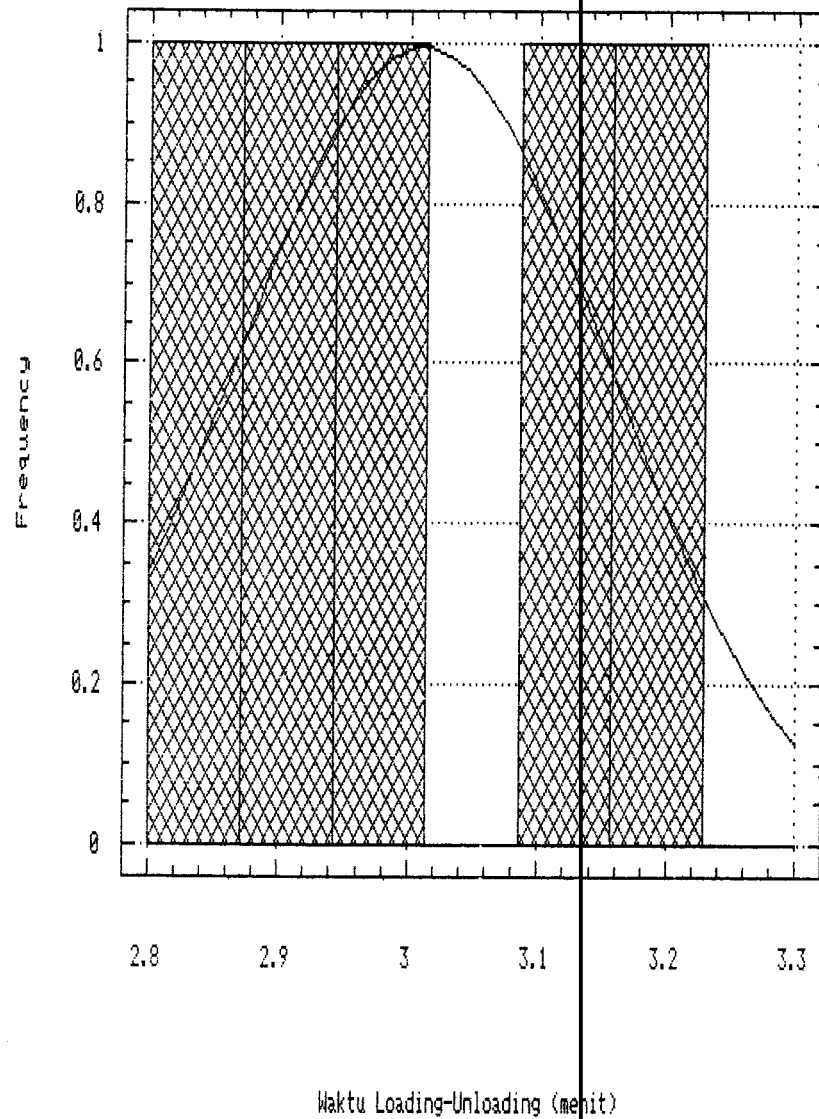
Frequency Histogram
Proses 1 Trash Bearing



Estimated KOLMOGOROV statistic DPLUS = 0.217509
 Estimated KOLMOGOROV statistic DMINUS = 0.244785
 Estimated overall statistic DN = 0.244785
 Approximate significance level = 0.925448

Frequency Histogram

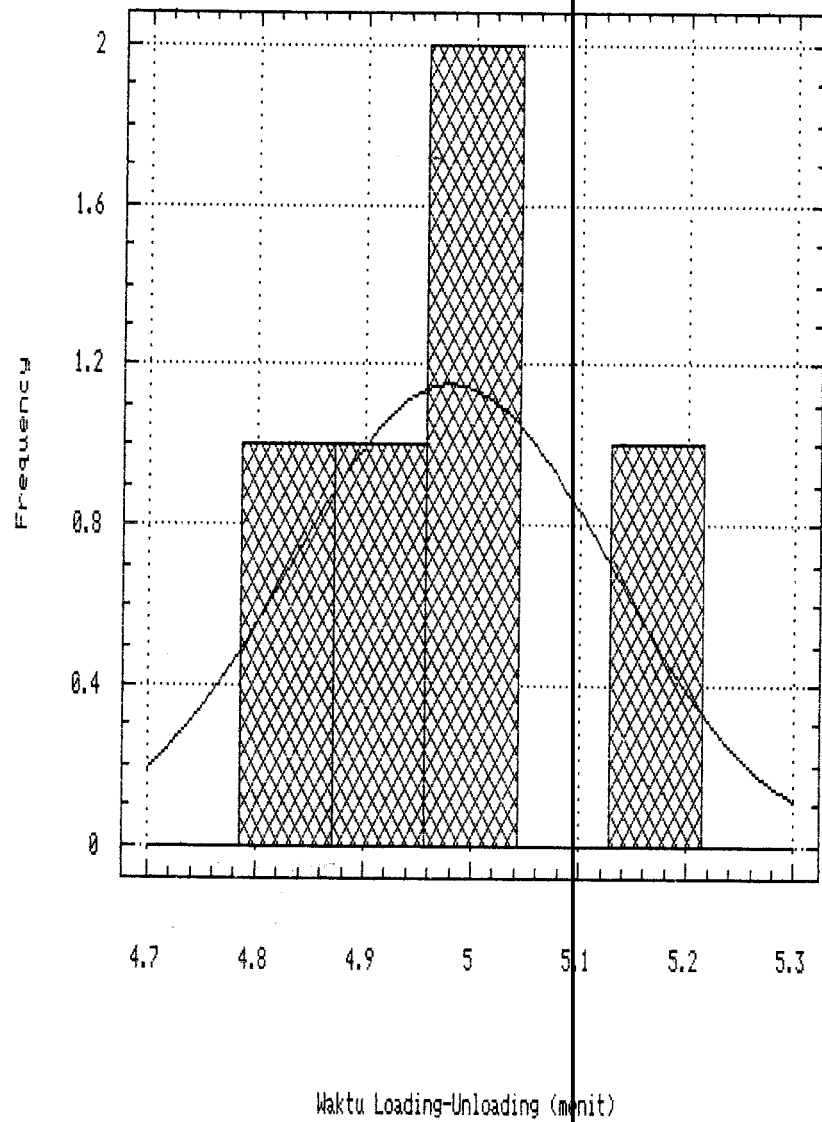
Proses 1 Counter Weight F.912



Estimated KOLMOGOROV statistic DPLUS = 0.178839
 Estimated KOLMOGOROV statistic DMINUS = 0.135192
 Estimated overall statistic DN = 0.178839
 Approximate significance level = 0.997203

Frequency Histogram

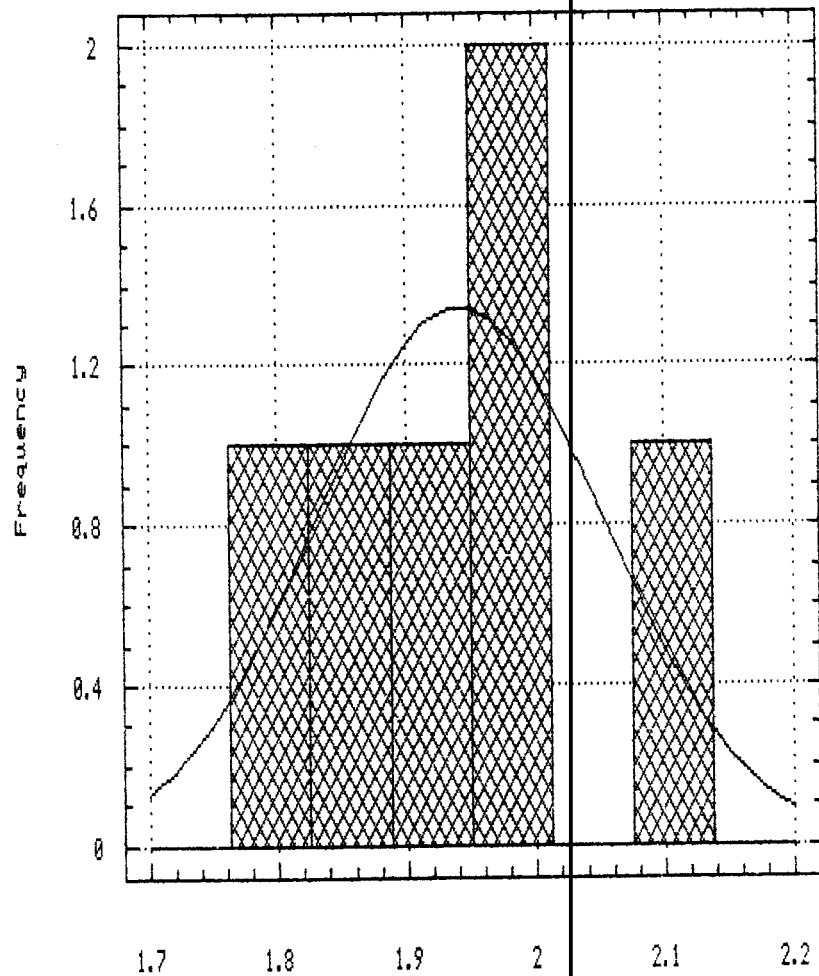
Proses 1 Bearing Cap



Estimated KOLMOGOROV statistic DPLUS = 0.246367
 Estimated KOLMOGOROV statistic DMINUS = 0.153633
 Estimated overall statistic DN = 0.246367
 Approximate significance level = 0.921917

Frequency Histogram

Proses 2 Exhaust Mazda 323

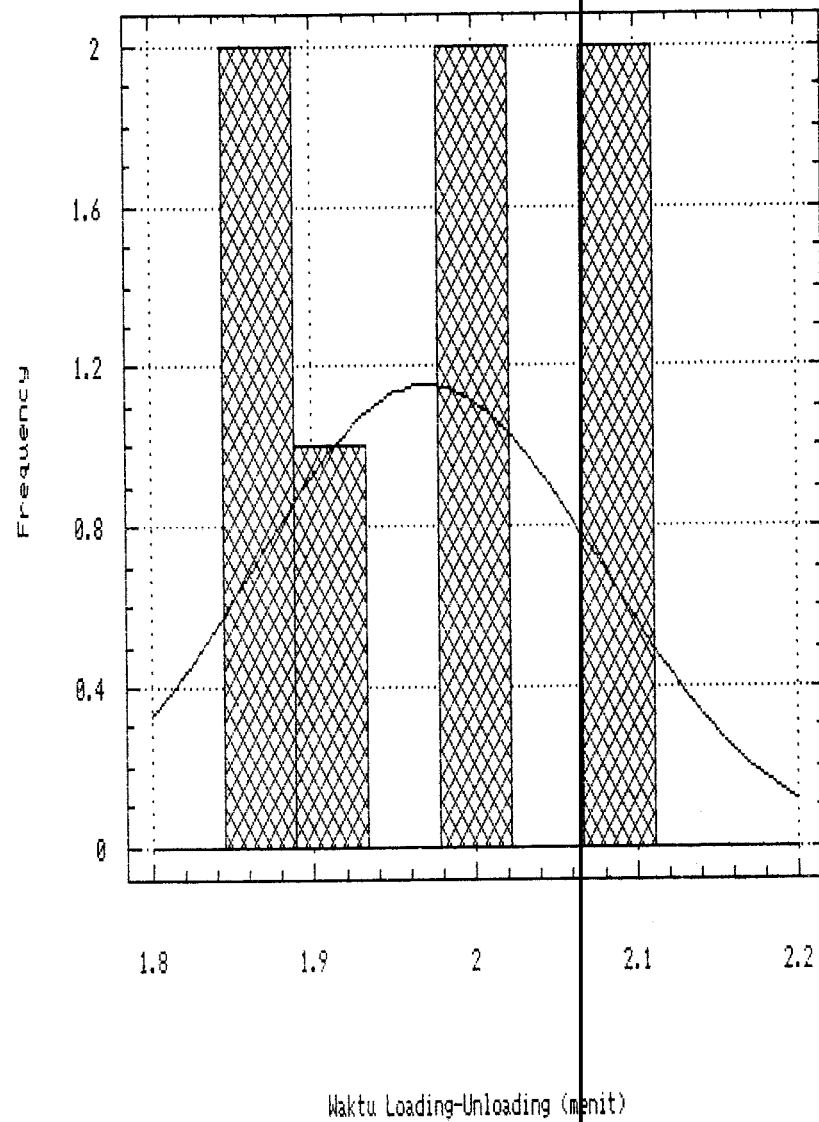


Waktu Loading-Unloading (menit)

Estimated KOLMOGOROV statistic DPLUS = 0.145784
 Estimated KOLMOGOROV statistic DMINUS = 0.199677
 Estimated overall statistic DN = 0.199677
 Approximate significance level = 0.970488

Frequency Histogram

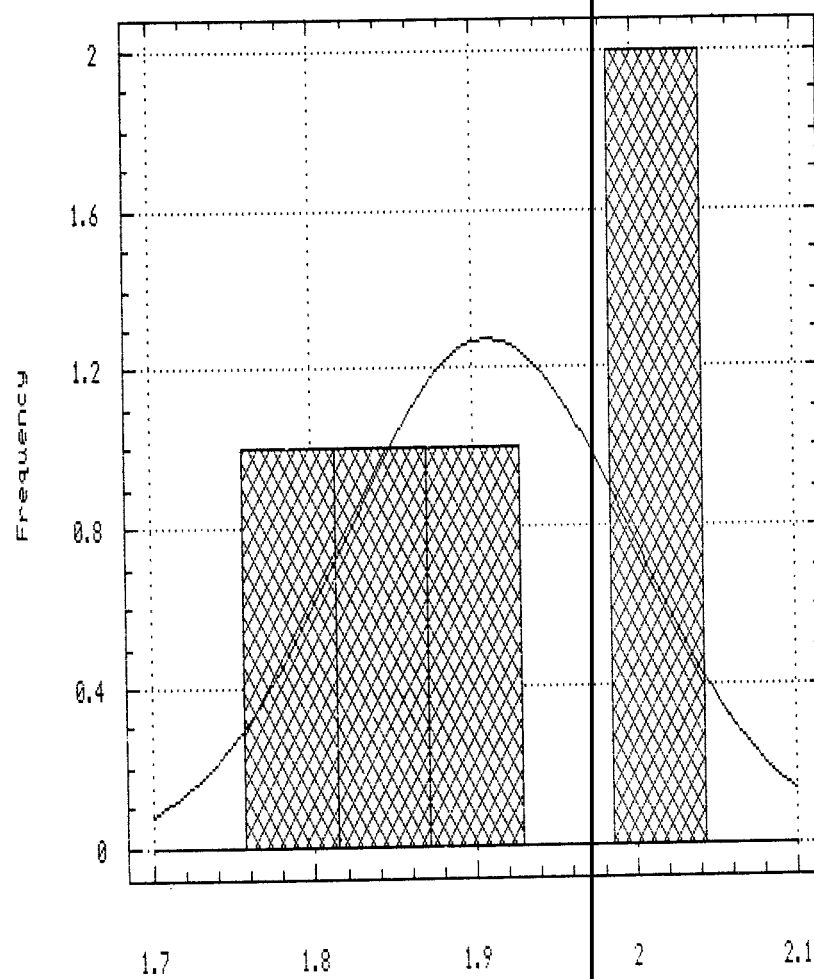
Proses 1 Exhaust Mazda 323



Estimated KOLMOGOROV statistic DPLUS = 0.175455
 Estimated KOLMOGOROV statistic DMINUS = 0.17626
 Estimated overall statistic DN = 0.17626
 Approximate significance level = 0.981521

Frequency Histogram

Proses 2 Intake Mazda 323

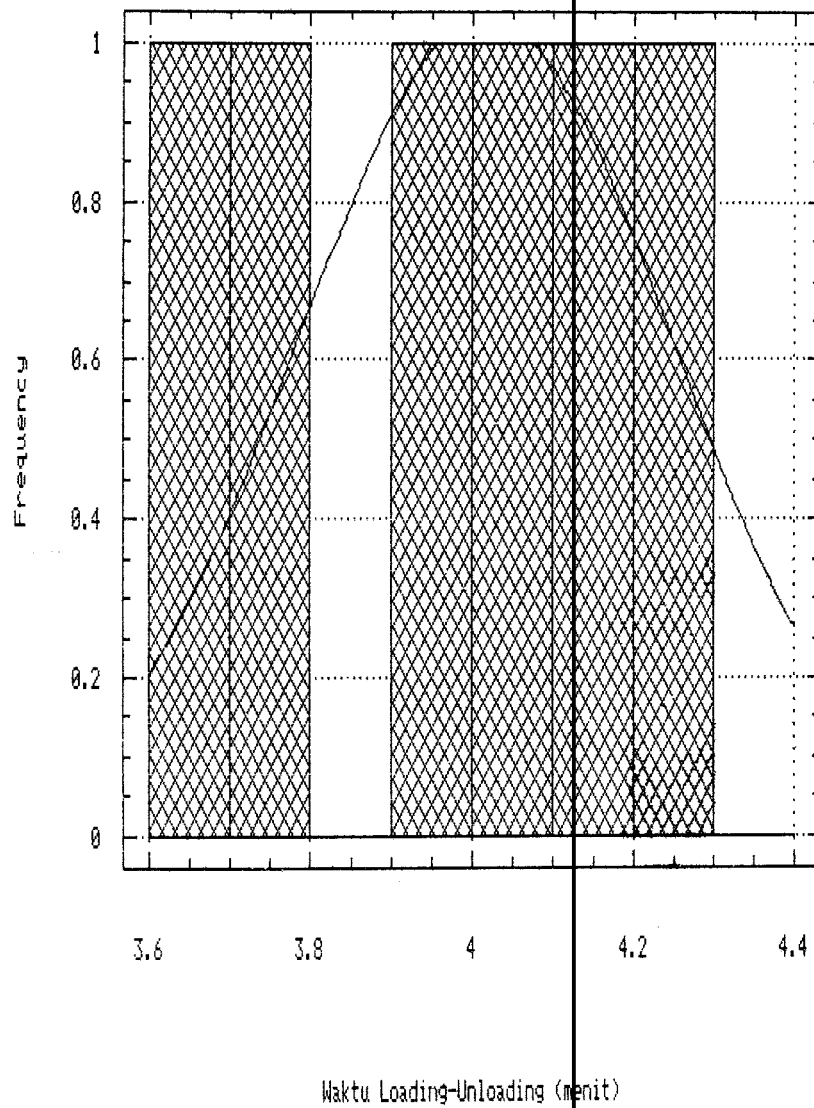


Waktu Loading-Unloading (menit)

Estimated KOLMOGOROV statistic DPLUS = 0.157152
 Estimated KOLMOGOROV statistic DMINUS = 0.242848
 Estimated overall statistic DN = 0.242848
 Approximate significance level = 0.929648

Frequency Histogram

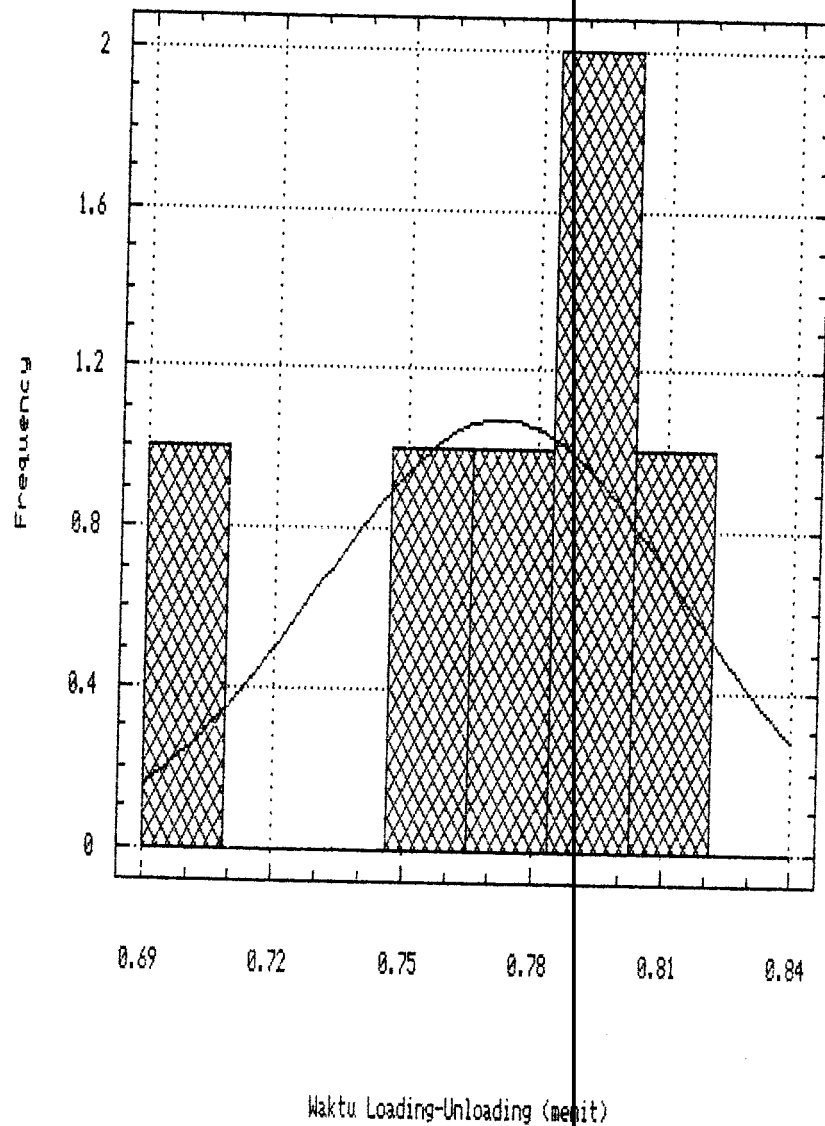
Proses 1 Intake Mazda 626



Estimated KOLMOGOROV statistic DPLUS = 0.158514
 Estimated KOLMOGOROV statistic DMINUS = 0.140469
 Estimated overall statistic DN = 0.158514
 Approximate significance level = 0.998197

Frequency Histogram

Proses 5 Intake Mazda 626



Estimated KOLMOGOROV statistic DPLUS = 0.178826
 Estimated KOLMOGOROV statistic DMINUS = 0.251712
 Estimated overall statistic DN = 0.251712
 Approximate significance level = 0.841616

Tabel 5.1
Waktu Loading-Unloading (menit)
Intake Mazda 626

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	4.2	4	3.8	3.7	4.1	4.3	
2	2	2.1	2.2	1.8	1.85	1.9	2
3	2.5	2.5	2.3	2.6	2.4		
4	2.4	2.7	2.6	2.6	2.5	2.3	
5	0.77	0.81	0.75	0.7	0.8	0.8	
6	0.57	0.58	0.53	0.55	0.61		
8	3.2	3.1	2.8	2.9	3	3	3.1

Tabel 5.2
Waktu Loading-Unloading (menit)
Intake Mazda 323

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	2	2	2.2	2.2	1.9	1.85	2.1
2	1.8	1.85	1.9	2	2		
3	2	2.2	2.3	2.1	1.9	2	2.1
4	0.85	0.9	0.8	0.85	0.9		
5	0.6	0.7	0.65	0.68	0.83	0.67	0.7
7	3.1	3.2	3.3	3	3.1	3.2	

Tabel 5.3
Waktu Loading-Unloading (menit)
Exhaust Mazda 323

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	2.1	2	1.85	1.9	1.85	2	2.1
2	2	1.8	1.9	2	1.85	2.1	
3	0.48	0.5	0.52	0.46	0.5	0.5	0.48
5	3.2	3	3.1	3	2.9		

Tabel 5.4
Waktu Loading-Unloading (menit)
Exhaust Mazda MR 90

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	3.2	3	2.9	2.8	3	3	2.8
2	3	2.9	2.8	2.75	3	3.1	
3	0.5	0.52	0.48	0.5	0.52	0.46	0.5
5	2.9	2.8	2.7	3	3.1		

Tabel 5.5
Waktu Loading-Unloading (menit)
Crank Case TS 230

No Proses	Pengukuran ke-							
	1	2	3	4	5	6	7	8
1	4.5	4.2	4.7	4.3	4.8	4.6	4.2	4.5
2	4.2	4.5	4.7	4.6	4.8	4.3		
3	4.8	4.5	4.3	4.4	4.5			
4	4.3	4.8	4.5	4.2	4.4			
5	10.5	10.5	10	9	9.5	10		

Tabel 5.6
Waktu Loading-Unloading (menit)
Crank Case TF 300

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	8.3	8	8.2	7.8	7.7		
2	7.5	7.8	8	8.3	8.2		
3	7.5	8.2	7.7	7.9	8		
4	8.4	8	7.6	7.8	8.2		
5	10.5	9.7	10	10.3	9.36		

Tabel 5.7
Waktu Loading-Unloading (menit)
Counter Weight F2L-912

No Proses	Pengukuran ke-						
	1	2	3	4	5	6	7
1	5.2	4.8	4.7	5	5.1	5.3	5
2	2.1	2	1.8	2	1.9	1.9	
3	1.75	1.85	2	2.1	1.9	2	
4	1.7	1.8	1.6	1.8	1.7	1.6	1.5

Tabel 5.8
Waktu Loading-Unloading
Counter Weight F3L-912

No Proses	Pengukuran ke-						
	1	2	3	4	5	6	7
1	5	5.3	4.8	5.1	5	4.9	4.7
2	1.9	1.8	2	2.1	1.8	1.9	
3	1.5	1.5	1.6	1.4	1.6	1.5	

Tabel 5.9
Waktu Loading-Unloading (menit)
Counter Weight F4L-912

No Proses	Pengukuran ke-						
	1	2	3	4	5	6	7
1	5.1	5.2	4.8	5	4.8		
2	2	1.9	1.8	2.1	2	1.8	2.1
3	1.6	1.8	1.7	1.7	1.6	1.8	1.9

Tabel 5.10
Waktu Loading-Unloading (menit)
Counter Weight F6L-912

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	5	4.5	4.8	4.7	5	5.2	
2	1.8	1.9	2	2.1	2	1.9	
3	1.8	1.9	2	1.7	1.8	1.9	

Tabel 5.11
Waktu Loading-Unloading (menit)
Cylinder Head FL-913

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	3.2	3	2.8	3.1	2.9	2.8	
2	0.5	0.52	0.48	0.46	0.54	0.5	0.48
3	4.5	5	4.7	5.1	5	4.8	
4	6.2	6	5.7	5.5	6		

Tabel 5.12
Waktu Loading-Unloading (menit)
Cylinder Head FL-912

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	3.1	2.85	2.9	3	3.2		
2	0.52	0.5	0.48	0.46	0.54	0.5	0.48
3	5.2	5	4.8	4.6	5		
4	6.2	6.3	5.8	6	5.6	6.1	

Tabel 5.13
Waktu Loading-Unloading (menit)
Bearing Cap

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	5	4.8	4.9	5	5.2		
2	4.5	4.8	5	4.7	5		
3	10.5	11	9.8	9.5	10		

Tabel 5.14
Waktu Loading-Unloading (menit)
Trash Bearing

No Proses	Pengukuran ke-						7
	1	2	3	4	5	6	
1	4.5	4.9	5	5.3	5		
2	4.8	5	4.6	5.2	5.1	5	
3	11	10.5	10	9.2	10.1		

Tabel 5.15
Waktu Pergantian Fixture (menit)

Ganti ke-	Pengukuran ke-						7
	1	2	3	4	5	6	
1	3	2.8	2.9	3.2	3.1		
2	2.1	2.2	2	1.9	1.8	2	2
3	1	0.9	0.95	1.1	1	1	
4	3.4	3	3.2	3.1	3		
5	1.08	0.94	1	1.1	1.05		
6	3.5	3.6	3.8	4	3.7		
7	2	2.1	1.9	2	2.1		
8	4.3	4	3.9	3.8	4		
9	3.7	4	4.3	3.9	4.1		

Tabel 5.16
Hasil Perhitungan Data
Intake Mazda 626

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn L	\bar{X}	Wn	Ws
1	6	4.44	0.21	4.65	3.38	0.159	0.998	4.02	4.02	4.623
2	7	6.96	0.13	2.37	1.59	0.154	0.996	1.98	1.98	2.277
3	5	2.46	0.1	2.77	2.15	0.237	0.941	2.46	2.583	2.9704
4	6	4.56	0.13	2.92	2.11	0.214	0.946	2.52	2.394	2.7531
5	6	3.35	0.04	0.88	0.67	0.252	0.842	0.77	0.847	0.9740
6	5	3.65	0.03	0.65	0.49	0.146	0.999	0.57	0.627	0.7210
8	7	2.73	0.12	3.39	2.64	0.172	0.986	3.01	3.01	3.4615

Tabel 5.17
Hasil Perhitungan Data
Intake Mazda 323

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	7	6.25	0.13	2.42	1.65	0.174	0.984	2.04	1.938	2.2287
2	5	2.81	0.08	1.67	1.91	0.243	0.93	1.91	2.101	2.4161
3	7	5.7	0.12	2.46	1.71	0.172	0.986	2.09	1.9855	2.2833
4	5	3.03	0.04	0.97	0.75	0.231	0.953	0.86	0.946	1.0879
5	7	4.33	0.03	0.76	0.56	0.163	0.993	0.66	0.726	0.8349
7	6	1.48	0.1	3.44	2.86	0.18	0.988	3.15	2.9925	3.4413

Tabel 5.18
Hasil Perhitungan Data
Exhaust Mazda 323

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	7	6	0.12	2.35	1.62	0.171	0.986	1.99	1.99	2.2885
2	6	4.39	0.1	2.25	1.64	0.2	0.97	1.94	2.134	2.4541
3	7	2.16	0.02	0.55	0.44	0.241	0.81	0.49	0.49	0.5635
5	5	1.8	0.1	3.35	2.73	0.237	0.94	3.04	2.888	3.3212

Tabel 5.19
Hasil Perhitungan Data
Exhaust Mazda MR 90

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{x}	Wn	Ws
1	7	3.48	0.14	3.39	2.56	0.16	0.994	2.97	2.97	3.4155
2	6	2.77	0.12	3.29	2.56	0.213	0.948	2.92	2.92	3.358
3	7	2.54	0.02	0.56	0.44	0.267	0.699	0.5	0.5	0.575
5	5	3.8	0.14	3.32	2.48	0.136	0.999	2.9	3.19	3.6685

Tabel 5.20
Hasil Perhitungan Data
Crank Case TS 230

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{x}	Wn	Ws
1	8	3.55	0.21	5.11	3.84	0.169	0.976	4.48	4.704	5.4096
2	6	3.51	0.21	5.15	3.88	0.159	0.998	4.52	4.294	4.9381
3	5	2.21	0.17	5	4	0.3	0.759	4.5	4.5	5.175
4	5	3.44	0.21	5.06	3.82	0.197	0.99	4.44	4.662	5.3613
5	6	4.63	0.53	11.52	8.32	0.223	0.926	9.92	10.416	11.978

Tabel 5.21
Hasil Perhitungan Data
Crank Case TF 300

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{x}	Wn	Ws
1	5	1.3	0.23	8.68	7.32	0.184	0.996	8	8	9.2
2	5	2.09	0.29	8.82	7.1	0.173	0.998	7.96	7.96	9.154
3	5	1.51	0.24	8.58	7.14	0.159	0.999	7.86	8.253	9.4909
4	5	2	0.28	8.85	7.15	0.136	0.999	8	8	9.2
5	5	1.87	0.34	11.05	8.99	0.198	0.99	10.02	10.02	11.523

Tabel 5.22
Hasil Perhitungan Data
Counter Weight F2L-912

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	7	2.44	0.2	5.6	4.43	0.187	0.967	5.01	5.01	5.7615
2	6	3.86	0.1	2.24	1.66	0.183	0.988	1.95	2.0475	2.3546
3	6	5.59	0.11	2.28	1.59	0.203	0.966	1.93	2.123	2.4414
4	7	6.08	0.1	1.98	1.67	0.173	0.985	1.67	1.837	2.1125

Tabel 5.23
Hasil Perhitungan Data
Counter Weight F3L-912

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	7	2.17	0.18	5.52	4.42	0.156	0.995	4.97	4.97	5.7155
2	7	4.57	0.1	2.24	1.62	0.223	0.926	1.93	2.0265	2.3304
3	6	3.28	0.07	1.72	1.31	0.254	0.833	1.52	1.672	1.9228

Tabel 5.24
Hasil Perhitungan Data
Counter Weight F4L-912

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	6	1.44	0.15	5.41	4.52	0.18	0.99	4.97	4.97	5.7155
2	7	5.8	0.12	2.31	1.6	0.203	0.934	1.96	2.058	2.3667
3	7	5.68	0.1	2.04	1.42	0.173	0.985	1.73	1.903	2.1884

Tabel 5.25
Hasil Perhitungan Data
Counter Weight F6L-912

Oprs	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	6	3.53	0.23	5.55	4.18	0.203	0.966	4.87	4.87	5.6005
2	6	3.86	0.1	5.24	1.66	0.183	0.988	1.95	2.0475	2.3546
3	6	4.29	0.1	2.14	1.56	0.183	0.988	1.85	2.035	2.3402

Tabel 5.26
Hasil Perhitungan Data
Cylinder Head FL-913

Oprs	N	N'	SD	BAK	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	6	4.04	0.15	3.14	5.52	0.179	0.99	2.97	2.97	3.4155
2	7	4.02	0.02	0.57	0.42	0.172	0.986	0.5	0.5	0.575
3	6	2.89	0.21	5.47	4.22	0.247	0.858	4.85	5.335	6.1352
4	5	2.85	0.25	6.62	5.14	0.267	0.867	5.88	5.586	6.4239

Tabel 5.27
Hasil Perhitungan
Cylinder Head FL-912

Oprs	N	N'	SD	BAK	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	5	2.9	0.13	3.39	2.63	0.179	0.997	3.01	3.01	3.4615
2	7	4.02	0.02	0.57	0.42	0.172	0.986	0.5	0.5	0.575
3	7	2.75	0.2	5.53	4.32	0.237	0.941	4.92	5.166	5.9409
4	6	2.52	0.24	6.71	5.29	0.167	0.996	6	5.7	6.555

Tabel 5.28
Hasil Perhitungan Data
Bearing Cap

Oprs	N	N'	SD	BAK	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	5	2.17	0.19	5.6	4.48	0.246	0.92	5.04	4.788	5.5062
2	5	2.5	0.19	5.37	4.23	0.227	0.959	4.8	5.28	6.072
3	5	4.38	0.53	11.75	8.57	0.206	0.984	10.16	9.652	11.099

Tabel 5.29
Hasil Perhitungan Data
Trash Bearing

Oprs	N	N'	SD	BAK	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	5	4.35	0.26	5.71	4.17	0.245	0.925	4.94	5.187	5.9650
2	6	2.56	0.2	5.54	4.36	0.258	0.82	4.95	4.95	5.6925
3	5	3.88	0.5	11.73	8.71	0.184	0.996	10.22	9.709	11.165

Tabel 5.30
Hasil Perhitungan Data
Pergantian Fixture

Ganti	N	N'	SD	BKA	BKB	Dn	Sgn	\bar{X}	Wn	Ws
1	5	3.56	0.14	3.42	2.58	0.136	0.999	3	3	3.45
2	7	5.71	0.12	2.36	1.64	0.214	0.905	2	2	2.3
3	6	5.99	0.06	1.17	0.81	0.283	0.721	0.99	0.99	1.1385
4	5	3.64	0.15	3.59	2.69	0.201	0.987	3.14	2.983	3.4304
5	7	4	0.05	1.18	0.87	0.241	0.811	1.02	1.02	1.173
6	5	3.42	0.17	4.24	3.2	0.141	0.999	3.72	3.348	3.8502
7	5	2.2	0.07	2.24	1.8	0.231	0.593	2.02	2.02	2.323
8	5	2.8	0.17	4.5	3.5	0.3	0.759	4	4	4.6
9	5	4	0.2	4.6	3.4	0.127	0.999	4	4	4.6

Kasus 1

Mesin = 4 Job = 5

Opr/job = 3

Matrik Routing

	1	2	3
1	1	2	0
2	1	4	2
3	1	2	3
4	1	3	4
5	3	4	0

Matrik Waktu Proses

	1	2	3
1	2	3	0
2	3	3	2
3	1	3	2
4	4	1	3
5	4	4	0

Kasus 2

Mesin = 5 Job = 5

Opr/job = 5

Matrik Routing

	1	2	3	4	5
1	1	2	3	4	5
2	2	3	4	5	1
3	3	4	5	1	2
4	4	5	1	2	3
5	5	1	2	3	4

Matrik Waktu Proses

	1	2	3	4	5
1	18	22	19	25	15
2	25	18	25	24	25
3	24	18	19	17	24
4	18	21	25	15	19
5	23	16	20	20	22

Kasus 3

Mesin = 5 Job = 6

Opr/job = 5

Matrik Routing

	1	2	3	4	5
1	1	2	3	4	5
2	1	3	5	2	4
3	3	1	2	5	4
4	3	4	5	1	2
5	4	2	1	5	3
6	5	4	3	2	1

Matrik Waktu Proses

	1	2	3	4	5
1	12	15	14	11	14
2	14	11	12	12	13
3	15	15	14	14	13
4	14	13	12	11	10
5	13	15	12	15	11
6	12	15	13	15	14

Kasus 4

Mesin = 5 Job = 8

Opr/job = 5

Matrik Routing

	1	2	3	4	5
1	1	2	3	4	5
2	5	4	3	2	1

3	1	2	3	5	4
4	2	5	3	4	1
5	5	3	1	2	4
6	3	4	5	1	2
7	5	4	2	3	1
8	3	2	1	5	4

Matrik Waktu Proses

	1	2	3	4	5
1	11	24	11	13	10
2	12	23	10	16	20
3	15	10	23	20	19
4	16	23	12	11	21
5	19	13	10	25	17
6	20	19	19	21	16
7	12	21	21	22	21
8	17	14	23	12	24

Kasus 5

Mesin = 5 Job = 10

Opr/job = 5

Matrik Routing

	1	2	3	4	5
1	1	2	3	4	5
2	5	4	3	2	1
3	2	1	3	4	5
4	3	4	5	2	1
5	2	3	4	5	1
6	5	2	1	4	3
7	2	4	5	1	3
8	3	1	4	5	2
9	2	4	5	1	3
10	1	5	4	2	3

Matrik Waktu Proses

	1	2	3	4	5
1	23	22	19	22	20
2	21	17	24	18	18
3	23	16	21	15	21
4	23	21	17	23	23
5	21	24	19	21	15
6	18	20	19	25	22
7	20	16	25	21	23
8	24	19	17	19	18
9	24	20	21	25	24
10	22	22	25	18	23

Kasus 6

Mesin = 5 Job = 12 Opr/job = 5

Matrik Routing

	1	2	3	4	5
1	5	4	3	2	1
2	2	3	4	5	1
3	4	5	1	2	3
4	2	3	1	5	4
5	2	1	4	3	5
6	5	4	2	1	3
7	3	1	2	5	4
8	1	5	3	4	2
9	2	4	1	3	5
10	1	2	3	4	5
11	5	4	3	2	1
12	2	3	4	5	1

Matrik Waktu Proses

	1	2	3	4	5
1	37	34	44	27	47
2	44	39	42	41	40
3	30	36	38	40	46

4	37	49	48	25	34
5	44	50	32	30	42
6	30	49	44	30	41
7	36	27	41	26	33
8	35	41	29	27	36
9	26	29	47	50	41
10	33	46	48	44	34
11	27	34	43	33	35
12	29	37	34	33	29

Kasus 7

Mesin = 5 Job = 15 Opr/job = 5

Matrik Routing

	1	2	3	4	5
1	4	5	3	2	1
2	1	2	3	5	4
3	2	4	5	1	3
4	2	3	4	5	1
5	1	4	5	2	3
6	1	3	4	5	2
7	2	1	3	4	5
8	5	4	2	1	3
9	2	1	4	5	3
10	5	4	3	2	1
11	4	2	1	3	5
12	4	2	1	5	3
13	3	2	1	4	5
14	4	3	1	2	4
15	2	5	3	4	1

Matrik Waktu Proses

	1	2	3	4	5
1	64	74	52	82	64
2	82	59	82	53	90
3	59	76	54	60	99
4	88	59	67	98	94
5	94	88	84	78	74
6	76	98	89	98	78
7	55	83	83	70	60
8	91	61	98	70	88
9	69	99	71	87	52
10	81	98	81	98	97
11	64	71	86	100	52
12	93	71	79	56	86
13	80	75	82	50	92
14	92	56	90	84	70
15	68	88	56	91	72

Kasus 8

Mesin = 5 Job = 20 Opr/job = 5

Matrik Routing

	1	2	3	4	5
1	1	2	3	4	5
2	5	4	3	2	1
3	2	3	1	5	4
4	4	5	3	2	1
5	5	4	3	2	1
6	3	2	1	4	5
7	4	5	2	1	3
8	2	3	1	4	5
9	1	2	4	5	3
10	3	1	2	4	5
11	2	1	3	4	5
12	2	4	5	1	3
13	3	1	5	2	4
14	3	4	2	1	5

15	1	2	3	4	5
16	2	3	4	5	1
17	2	1	3	4	5
18	4	5	1	2	3
19	3	2	1	5	4
20	2	3	1	4	5

Matrik Waktu Proses

	1	2	3	4	5
1	72	78	86	29	27
2	54	28	41	43	68
3	69	34	59	44	28
4	64	60	65	49	83
5	93	36	36	52	43
6	97	97	26	63	93
7	57	74	60	80	69
8	75	59	41	56	56
9	62	68	27	46	84
10	74	69	26	49	36
11	76	80	91	82	67
12	59	68	37	85	67
13	98	89	58	83	56
14	59	47	25	78	39
15	73	91	48	39	90
16	85	91	42	90	70
17	58	76	91	81	74
18	70	27	59	80	37
19	74	27	61	68	94
20	88	69	37	62	59

Kasus 9

Mesin = 6 Job = 5

Opr/job = 6

Matrik Routing

	1	2	3	4	5	6
1	6	5	4	3	2	1
2	1	2	3	4	5	6
3	2	4	5	1	3	6
4	2	4	5	1	3	6
5	2	6	4	5	1	3

Matrik Waktu Proses

	1	2	3	4	5	6
1	8	8	7	5	5	10
2	7	5	6	5	9	6
3	10	9	9	8	8	8
4	8	10	8	9	7	9
5	6	6	9	6	8	8

Kasus 10

Mesin = 6 Job = 12 Opr/job = 6

Matrik Routing

	1	2	3	4	5	6
1	2	6	4	5	1	3
2	3	4	5	6	2	1
3	1	5	4	2	3	6
4	5	1	2	4	3	6
5	5	4	2	1	3	6
6	3	2	1	6	5	4
7	4	1	5	2	6	3
8	6	5	4	2	3	1
9	1	2	4	5	3	6
10	6	2	5	3	4	1
11	1	5	2	4	3	6
12	6	5	4	3	2	1

Matrik Waktu Proses

	1	2	3	4	5	6
1	23	24	24	22	20	10
2	24	11	18	13	12	19
3	16	16	15	15	16	17
4	19	15	14	17	23	24
5	24	15	17	11	19	15
6	22	14	22	15	18	14
7	19	11	25	18	20	19
8	24	14	24	16	12	10
9	16	21	12	15	19	11
10	18	16	25	25	20	22
11	15	23	14	22	17	12
12	13	11	21	13	21	17

Kasus 11

Mesin = 10

Job = 5

Opr/job = 8

Matrik Routing

	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	8	7	6	5	4	3	2	1
3	2	4	5	1	6	3	8	7
4	8	7	4	5	6	1	2	3
5	3	2	1	4	5	6	7	8

Matrik Waktu Proses

	1	2	3	4	5	6	7	8
1	9	7	6	10	10	5	8	7
2	7	8	8	9	8	8	8	7
3	10	5	6	9	5	10	10	6
4	10	8	9	7	8	9	10	5
5	7	10	5	10	10	8	9	7

Kasus 12

Mesin = 10

Job = 6

Opr/job = 8

Matrik Routing

	1	2	3	4	5	6	7	8
1	8	4	7	5	6	1	2	3
2	3	2	1	4	7	5	6	8
3	8	7	6	5	4	3	2	1
4	2	4	6	8	1	3	5	7
5	3	2	4	5	7	8	6	1
6	1	2	3	4	5	6	7	8

Matrik Waktu Proses

	1	2	3	4	5	6	7	8
1	12	15	13	13	22	21	14	12
2	12	22	11	23	11	16	10	14
3	13	14	17	20	14	22	15	16
4	17	19	18	20	21	22	16	20
5	14	15	25	19	22	18	19	15
6	20	23	21	20	21	22	19	21

Kasus 13

Mesin = 10

Job = 8

Opr/job = 8

Matrik Routing

	1	2	3	4	5	6	7	8
1	5	4	7	8	3	2	1	6
2	2	1	3	4	5	6	7	8
3	8	7	6	5	4	3	2	1
4	2	1	3	4	5	6	7	8
5	1	8	7	5	4	3	2	6
6	2	4	6	8	1	3	5	7
7	1	2	3	6	5	4	8	7
8	6	5	4	7	8	1	2	3

Matrik Waktu Proses

	1	2	3	4	5	6	7	8
1	13	13	10	25	23	24	14	21
2	19	10	21	14	11	11	18	15
3	12	25	20	22	16	16	16	16
4	13	13	24	15	22	14	20	14
5	13	14	25	12	21	21	13	15
6	11	13	12	20	21	14	19	15
7	16	18	21	23	21	14	10	11
8	25	15	14	17	17	19	18	17

Kasus 14

Mesin = 10

Job = 6

Opr/job = 10

Matrik Routing

	1	2	3	4	5	6	7	8	9	10
1	10	9	8	7	6	5	4	3	2	1
2	2	3	1	4	5	8	7	9	10	6
3	5	6	7	8	9	10	1	2	3	4
4	2	4	5	1	6	3	7	8	9	10
5	2	10	8	9	7	4	5	3	6	1
6	8	6	4	2	1	9	3	5	10	7

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10
1	12	84	6	19	90	72	35	92	61	29
2	40	25	7	45	50	38	98	18	31	38
3	14	51	78	25	36	74	41	77	27	25
4	76	36	97	99	69	96	84	63	89	78
5	16	28	17	72	90	41	69	49	41	22
6	66	10	13	85	82	55	94	49	54	60

Kasus 15

Mesin = 10

Job = 8

Opr/job = 10

Matrik Routing

	1	2	3	4	5	6	7	8	9	10
1	10	1	3	4	6	2	5	7	9	8
2	2	4	5	1	6	7	9	8	10	3
3	3	4	6	9	7	8	2	10	1	5
4	5	9	7	3	1	6	4	8	10	2
5	2	8	7	9	6	4	3	1	10	5
6	9	8	7	6	5	4	3	2	1	10
7	2	4	5	1	3	6	8	7	9	10
8	5	6	7	8	9	10	1	2	3	4

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10
1	47	55	93	79	6	91	24	71	31	90
2	36	77	43	29	96	9	97	88	73	88
3	63	11	90	30	76	9	72	27	12	72
4	67	42	57	26	14	99	93	12	95	20
5	8	37	87	62	20	53	94	30	75	79
6	46	52	87	24	56	50	25	22	20	54
7	16	37	95	97	97	46	28	9	88	23
8	29	53	69	99	44	58	76	37	25	58

Kasus 16

Mesin = 10

Job = 10

Opr/job = 10

Matrik Routing

	1	2	3	4	5	6	7	8	9	10
1	1	3	5	7	9	2	4	6	8	10
2	1	2	3	4	5	6	7	8	9	10
3	2	3	4	5	6	7	8	9	10	1
4	2	4	6	8	10	1	3	5	7	9
5	3	4	5	6	7	8	9	10	1	3
6	4	5	6	7	8	9	10	1	3	2
7	5	6	7	8	9	10	1	2	3	4

8	10	8	9	7	6	5	4	2	3	1
9	8	7	6	4	5	2	1	3	10	9
10	7	8	9	10	1	2	3	4	5	6

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10
1	31	32	29	25	34	29	26	30	25	25
2	31	31	25	33	34	25	32	27	30	33
3	25	29	27	30	27	31	35	28	35	34
4	34	31	29	27	34	34	34	35	32	27
5	31	25	28	25	33	29	31	35	26	29
6	25	30	34	31	28	29	25	30	26	35
7	30	33	33	29	26	30	32	31	29	26
8	25	34	30	29	34	31	31	29	31	30
9	35	32	34	28	29	27	27	31	27	26
10	27	34	32	30	26	30	34	26	32	30

Kasus 17

Mesin = 10

Job = 12 Opr/job = 10

Matrik Routing

	1	2	3	4	5	6	7	8	9	10
1	10	9	8	7	6	5	4	3	2	1
2	1	3	5	7	9	2	4	6	8	10
3	2	4	6	8	10	9	7	5	3	1
4	1	2	3	4	5	6	7	8	9	10
5	6	7	8	9	10	1	2	3	4	5
6	7	8	9	10	1	2	3	4	5	6
7	2	1	4	3	6	5	8	7	10	9
8	7	6	9	8	1	10	3	2	5	4
9	8	9	10	1	2	3	4	5	6	7
10	9	10	7	8	5	6	3	4	1	2
11	5	6	7	8	9	10	1	2	3	4
12	10	9	8	7	6	1	2	3	4	5

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10
1	18	20	28	29	17	71	75	29	48	94
2	31	32	17	22	83	74	21	88	92	40
3	69	44	97	27	58	78	64	69	15	96
4	17	51	94	94	58	70	40	30	64	84
5	40	10	30	97	31	47	60	73	33	27
6	42	22	49	68	16	85	27	26	49	90
7	80	62	70	86	88	89	83	57	59	59
8	59	20	88	42	71	12	60	35	38	74
9	68	93	16	62	76	5	50	29	85	94
10	58	73	24	94	34	15	89	26	30	46
11	45	20	66	12	99	7	73	61	74	58
12	71	76	84	91	49	80	50	67	59	34

Kasus 18

Mesin = 10

Job = 15 Opr/job = 10

Matrik Routing

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	10	9	8	7	6
2	10	9	8	7	6	1	2	3	4	5
3	2	4	6	8	10	1	3	5	7	9
4	4	6	8	10	9	7	5	3	1	2
5	6	7	8	9	10	5	4	3	2	1
6	1	2	3	4	5	6	7	8	9	10
7	5	4	3	2	1	6	7	8	9	10
8	9	10	1	2	3	4	5	6	7	8
9	3	4	5	6	7	8	9	10	1	2
10	2	1	4	3	6	5	8	7	10	9
11	7	8	9	10	1	2	3	4	5	6
12	10	6	5	4	2	3	7	8	9	1
13	5	6	8	4	3	1	2	7	9	10
14	6	4	9	10	3	5	7	8	2	1
15	5	6	7	8	9	10	1	2	3	4

Matrik waktu Proses

	1	2	3	4	5	6	7	8	9	10
1	57	85	28	59	72	68	84	48	44	47
2	91	48	83	86	94	35	67	33	41	88
3	50	90	97	71	34	83	54	61	60	45
4	82	52	81	66	81	84	88	72	75	26
5	33	94	74	68	68	93	79	34	94	36
6	78	40	56	88	94	70	86	84	81	88
7	25	86	94	97	83	38	33	70	41	58
8	67	96	100	74	38	98	92	75	38	69
9	95	78	89	83	84	51	83	52	56	55
10	27	93	44	91	27	67	46	91	81	48
11	33	84	37	57	55	53	43	70	96	28
12	58	76	83	61	62	35	33	44	84	70
13	76	44	29	45	30	90	92	46	51	88
14	84	61	85	49	29	76	65	93	73	26
15	33	61	56	82	77	39	46	55	83	41

Kasus 19

Mesin = 15

Job = 10 Opr/job = 15

Matrik Routing

	1	2	3	4	5	6	7	8	9	10	11	12
1	15	14	13	12	11	10	9	8	7	6	5	4
3	2	4	6	8	10	12	14	1	3	5	7	9
3	10	11	12	13	14	15	1	2	3	4	5	6
4	1	2	3	4	5	6	7	8	9	10	11	12
5	2	1	4	3	6	5	8	7	10	9	12	11
6	8	9	10	11	12	13	14	15	1	2	3	4
7	7	8	9	10	11	12	13	14	15	1	2	3
8	5	6	7	8	9	10	11	12	13	14	15	1
9	9	10	11	12	13	14	15	8	7	6	5	4
10	15	13	11	9	7	5	1	2	4	6	8	10
	13	14	15									
1	3	2	1									
2	11	13	15									
3	7	8	9									
4	13	14	15									
5	15	13	14									
6	5	6	7									
7	4	5	6									
8	2	3	4									
9	3	2	1									
10	12	14	3									

Matrik Waktu Proses

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1	32	33	45	30	38	32	44	47	41	34	34	37
2	26	41	35	39	42	34	37	33	44	25	36	31
3	30	28	42	34	39	34	45	39	45	30	40	31
4	49	33	38	47	40	41	32	31	45	25	42	29
5	46	41	43	30	44	33	41	25	41	50	43	38
6	42	40	30	34	28	34	34	42	36	33	27	38
7	27	50	28	38	47	40	40	27	32	30	44	25
8	45	26	28	37	25	31	27	41	25	25	41	29
9	41	38	37	30	36	36	47	37	38	46	43	50
10	45	49	32	26	32	32	46	44	50	41	34	50

	13	14	15
1	35	47	38
2	40	50	42
3	50	50	32
4	39	48	28
5	28	43	45
6	34	29	50
7	46	50	29
8	36	41	50
9	33	37	47
10	41	29	43

Kasus 20

Mesin = 15

Job = 8

Opr/job = 15

Matrik Routing

	1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	1	4	5	6	8	7	9	10	11	12
2	15	14	13	12	11	10	9	8	7	6	5	4
3	12	13	14	15	1	2	3	4	5	6	7	8
4	10	11	12	13	14	15	1	3	5	7	9	8
5	8	9	10	11	12	13	14	15	7	5	1	2
6	1	2	3	4	5	6	7	8	9	10	11	12
7	4	5	6	7	8	9	10	11	12	13	14	15
8	11	12	13	14	15	2	4	6	8	10	9	7

	13	14	15
1	13	14	15
2	3	2	1
3	9	10	11
4	6	4	2
5	4	6	3
6	13	14	15
7	1	2	3
8	1	3	5

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10	11	12
1	97	86	97	73	63	68	26	99	29	92	57	59
2	57	62	92	66	73	62	26	36	47	98	46	48
3	42	44	98	54	58	68	78	53	68	87	67	39
4	39	25	88	61	28	50	47	61	38	61	55	37
5	90	36	86	80	44	87	83	41	31	70	78	71
6	41	34	65	56	28	70	25	89	28	54	28	73
7	97	37	53	33	27	82	85	50	87	81	72	75
8	80	98	70	42	53	45	63	50	26	49	75	40

	13	14	15
1	34	26	74
2	54	34	44
3	85	88	68
4	43	37	45
5	97	47	29
6	41	75	52
7	98	40	89

Kasus 21

Mesin = 20

Job = 5

Opr/job = 20

Matrik Routing

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	20	19	18	17	16	15	14	13	12	11	10	9
3	2	3	4	5	6	7	8	9	10	11	12	13
4	20	10	19	2	18	3	17	4	16	5	15	6
5	5	6	7	8	9	10	15	16	17	18	19	20

	13	14	15	16	17	18	19	20
1	13	14	15	16	17	18	19	20
2	8	7	6	5	4	3	2	1
3	14	15	16	17	18	19	20	1
4	14	7	13	8	12	9	13	11
5	1	2	3	4	11	12	13	14

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10	11	12
1	52	32	37	53	43	31	47	54	49	35	38	54
2	42	41	40	56	43	50	37	41	37	60	37	43
3	53	53	31	36	46	54	51	37	58	54	59	56
4	47	60	39	59	53	34	47	30	31	37	58	45
5	34	60	37	53	43	55	46	52	33	59	60	41

	13	14	15	16	17	18	19	20
1	42	30	58	59	31	60	50	36
2	32	30	43	40	31	36	49	31
3	48	41	38	42	60	60	50	42
4	57	48	37	60	47	48	48	37
5	34	46	36	50	37	36	59	60

Kasus 22

Mesin = 20

Job = 10

Opr/job = 20

Matrik Routing

	1	2	3	4	5	6	7	8	9	10	11	12
1	3	6	9	12	15	18	1	2	4	5	7	8
2	2	4	6	8	10	12	14	16	18	20	1	3
3	2	3	5	7	11	13	17	19	1	4	6	8
4	20	19	18	17	16	15	14	13	12	11	10	9
5	1	2	3	4	5	6	7	8	9	10	20	19
14	15	16	17	18	19	20	14	13	12	11	10	1
7	2	1	4	3	6	5	8	7	10	9	12	11
8	9	10	11	6	7	8	1	2	3	5	4	16
9	10	11	12	13	14	15	16	17	18	19	20	1
10	1	5	2	3	4	6	7	8	10	9	20	16

	13	14	15	16	17	18	19	20
1	10	11	16	17	19	20	13	14
2	5	7	9	11	13	15	17	19
3	9	10	12	14	15	16	18	20
4	8	7	6	5	4	3	2	1
5	18	17	16	15	14	13	12	11
6	2	3	4	5	6	7	8	9
7	14	13	16	15	18	17	20	19
8	20	19	18	17	12	13	14	15
9	2	3	4	5	6	7	8	9
10	19	18	17	11	12	13	14	15

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10	11	12
1	47	50	32	42	35	52	35	39	36	37	51	48
2	50	49	58	36	52	54	31	35	47	51	48	39
3	43	40	33	38	48	37	47	38	40	56	40	46
4	56	32	55	45	44	36	50	41	42	43	53	49
5	47	38	40	32	46	43	39	38	35	33	42	52
6	56	43	30	42	39	42	34	44	37	32	54	42

7	46	57	46	50	57	35	58	30	45	44	53	32
8	34	60	42	30	44	40	38	53	37	50	44	36
9	34	33	57	37	53	32	33	43	59	59	39	34
10	44	55	56	36	34	57	47	47	49	40	39	45

	13	14	15	16	17	18	19	20
1	38	50	42	47	37	54	54	39
2	35	48	33	34	51	55	55	39
3	45	32	30	35	49	51	57	44
4	42	60	54	34	37	42	30	59
5	49	35	37	33	49	31	59	54
6	53	41	44	31	51	46	59	30
7	34	50	41	31	48	43	33	57
8	40	51	39	54	49	40	48	52
9	30	40	39	57	31	31	60	53
10	30	38	30	49	56	36	44	34

Kasus 23

Mesin = 20

Job = 15 Opr/job = 20

Matrik Routing

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	3	4	5	6	7	8	9	10	11	12	13
3	3	4	5	6	7	8	9	10	11	12	13	14
4	4	5	6	7	8	9	10	11	12	13	14	15
5	10	11	12	13	14	15	16	17	18	19	20	1
6	15	16	17	18	19	20	1	2	3	4	5	6
7	16	14	12	10	8	6	4	2	1	3	5	7
8	1	3	5	7	9	11	13	15	17	19	2	4
9	20	19	18	17	16	15	14	13	12	11	1	2
10	1	2	3	4	8	7	6	5	9	10	11	12
11	4	5	6	7	8	9	10	1	2	3	15	14
12	20	19	18	17	16	15	14	13	12	11	10	9
13	6	7	8	9	1	2	3	4	12	11	10	9
14	12	13	14	15	16	1	2	3	4	5	20	19
15	1	2	3	4	5	6	7	8	9	10	11	12

	13	14	15	16	17	18	19	20
1	13	14	15	16	17	18	19	20
2	14	15	16	17	18	19	20	1
3	15	16	17	18	19	20	1	2
4	16	17	18	19	20	1	2	3
5	2	3	4	5	6	7	8	9
6	7	8	9	10	11	12	13	14
7	9	11	13	15	17	19	18	20
8	6	8	10	12	14	16	18	20
9	3	4	5	6	7	8	9	10
10	16	15	14	13	17	18	19	20
11	13	12	11	20	19	18	17	16
12	8	7	6	5	4	3	2	1
13	20	19	18	17	16	15	14	13
14	18	17	16	15	6	7	8	9
15	13	14	15	16	17	18	19	20

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10	11	12
1	51	34	51	42	60	50	33	55	57	56	43	40
2	53	30	35	54	43	46	58	59	57	42	49	55
3	55	43	32	35	48	43	43	47	32	47	38	35
4	53	37	31	31	45	37	54	40	33	38	58	57
5	31	50	59	30	54	59	35	37	53	33	31	55
6	48	35	49	50	42	59	54	33	39	58	57	43
7	44	56	57	31	47	50	33	59	59	34	50	53
8	55	56	51	46	56	43	30	34	36	39	53	43
9	30	31	50	38	52	58	44	48	37	59	45	38
10	50	50	33	40	37	43	43	58	48	44	58	53
11	35	42	57	39	39	36	60	46	60	34	40	60
12	57	46	52	35	50	49	40	35	50	50	33	32
13	44	46	52	37	52	52	34	59	43	32	46	52
14	58	56	35	37	50	59	58	30	35	56	40	54
15	32	45	44	51	50	34	57	58	50	34	40	37

	13	14	15	16	17	18	19	20
1	46	57	59	36	50	52	58	40
2	60	52	55	38	37	43	55	32
3	44	57	44	43	52	35	43	57
4	54	37	52	51	32	41	47	46
5	39	58	49	53	42	33	33	56
6	31	30	39	31	58	56	52	55
7	53	34	46	41	54	43	55	54
8	57	56	37	39	47	44	53	47
9	44	40	30	58	60	59	48	56
10	37	46	34	41	32	38	46	53
11	49	42	47	45	52	37	44	55
12	55	31	34	59	50	36	59	60
13	49	53	36	36	50	34	39	39
14	37	34	30	41	52	58	60	42
15	49	31	36	47	50	48	53	47

Kasus 24

Mesin = 20

Job = 20 Opr/job = 5

Matrik Routing

	1	2	3	4	5
1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	16	17	18	19	20
5	20	19	18	17	16
6	15	14	13	12	11
7	10	9	8	7	6
8	5	4	3	2	1
9	2	4	8	10	12
10	14	16	18	20	1
11	1	3	5	7	9
12	11	13	15	17	19
13	4	6	8	10	12
14	10	11	12	13	15
15	16	17	18	19	20
16	8	10	12	14	16
17	10	12	14	16	18
18	5	6	7	8	9
19	15	16	17	18	20

20 | 20 19 18 17 16

Matrik Waktu Proses

	1	2	3	4	5
1	72	78	86	29	27
2	54	28	41	43	68
3	69	34	59	44	28
4	64	60	65	49	83
5	93	36	36	52	43
6	97	97	26	63	93
7	57	74	60	80	69
8	75	59	41	56	56
9	62	68	27	46	84
10	74	69	26	49	36
11	76	80	91	82	67
12	59	68	37	85	67
13	98	89	58	83	56
14	59	47	25	78	39
15	73	91	48	39	90
16	85	91	42	90	70
17	58	76	91	81	74
18	70	27	59	80	37
19	74	27	61	68	94
20	88	69	37	62	59

Kasus 25 (PT. BBI Surabaya)

Mesin = 25

Job = 14 Opr/job = 8

Matrik Routing

	1	2	3	4	5	6	7	8
1	1	2	1	1	3	3	4	5
2	6	6	1	7	7	4	5	0
3	8	8	7	4	5	0	0	0
4	9	10	7	4	5	0	0	0
5	1	12	13	14	5	0	0	0
6	15	16	17	18	5	0	0	0
7	19	20	21	22	23	0	0	0
8	19	20	21	23	0	0	0	0
9	19	20	21	23	0	0	0	0
10	19	20	21	23	0	0	0	0
11	6	24	6	5	0	0	0	0
12	6	24	6	5	0	0	0	0
13	19	25	5	0	0	0	0	0
14	19	25	5	0	0	0	0	0

Matrik Waktu Proses

	1	2	3	4	5	6	7	8
1	14.62	22.28	21.82	17.75	10.97	6.72	5.00	0.89
2	16.31	32.22	33.14	8.12	7.77	6.67	1.20	0.00
3	23.05	16.60	8.58	2.67	1.18	0.00	0.00	0.00
4	35.13	35.04	4.74	4.50	2.73	0.00	0.00	0.00
5	30.34	42.45	40.15	42.80	5.24	0.00	0.00	0.00
6	27.47	40.10	40.99	37.47	4.72	0.00	0.00	0.00
7	2.62	10.92	17.73	7.85	3.33	0.00	0.00	0.00
8	3.39	15.76	21.38	4.00	0.00	0.00	0.00	0.00
9	2.46	12.91	16.73	3.00	0.00	0.00	0.00	0.00
10	2.20	11.58	16.07	2.83	0.00	0.00	0.00	0.00
11	21.02	2.19	17.36	0.81	0.00	0.00	0.00	0.00
12	57.21	5.96	38.85	2.21	0.00	0.00	0.00	0.00
13	1.81	46.45	0.76	0.00	0.00	0.00	0.00	0.00
14	2.50	20.50	0.60	0.00	0.00	0.00	0.00	0.00

Kasus 26

Mesin = 3 Job = 5

Opr/job = 10

Matrik Routing

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	3	2	1	1	2	3	1
2	2	1	2	3	2	1	3	1	2	1
3	3	2	1	2	1	3	2	1	3	1
4	3	2	1	1	2	3	1	2	3	2
5	2	3	1	2	1	3	3	2	1	2

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10
1	10	24	25	21	20	15	23	23	20	22
2	26	30	29	24	26	30	30	25	25	26
3	21	16	15	14	19	18	24	21	21	12
4	25	23	27	28	23	26	24	25	26	21
5	24	28	26	24	23	20	22	33	25	21

Kasus 27

Mesin = 3 Job = 8

Opr/job = 15

Matrik Routing

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	3	2	1	1	2	3	2	1	2	1	3	1	2
2	1	2	3	2	3	1	2	1	3	3	1	2	1
3	2	1	3	1	2	3	3	1	2	2	3	1	2
4	2	1	2	1	2	1	3	2	2	3	3	2	1
5	1	3	3	1	2	3	2	1	3	1	2	3	1
6	2	1	3	1	2	3	2	1	2	1	3	2	1
7	2	3	1	2	2	1	3	2	1	3	2	1	2
8	1	2	3	2	1	2	1	3	3	1	2	3	2

	14	15
1	3	1
2	2	3
3	1	3
4	2	1
5	1	3
6	3	2
7	1	3
8	1	3

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	45	52	55	49	51	52	20	45	56	57	46	48	35
2	40	43	45	44	55	66	52	41	33	25	54	55	24
3	20	23	24	53	24	21	34	25	65	46	21	22	44
4	47	40	30	24	35	26	24	34	42	43	46	52	42
5	65	64	55	58	57	60	61	44	46	57	48	59	67
6	50	30	24	25	26	34	65	56	42	23	54	56	57
7	36	64	65	35	45	56	45	46	50	40	54	50	56
8	60	45	54	50	52	50	60	45	54	57	45	54	25

	14	15
1	35	46
2	50	43
3	23	24
4	40	47
5	55	64
6	45	30
7	45	64
8	64	54

Kasus 28

Mesin = 4 Job = 5

Opr/job = 10

Matrik Routing

	1	2	3	4	5	6	7	8	9	10
1	2	4	3	1	2	4	1	3	4	3
2	1	2	3	4	4	3	2	1	2	1
3	3	2	1	4	2	1	3	4	1	2
4	2	3	4	2	3	1	2	4	1	2
5	4	1	3	2	1	3	3	2	1	4

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10
1	54	45	50	30	34	45	51	65	42	43
2	45	65	55	50	54	35	50	52	50	52
3	34	33	36	34	54	44	55	50	50	64
4	45	52	55	66	32	30	34	53	46	54
5	62	52	45	45	51	51	52	50	50	34

Kasus 29

Mesin = 5 Job = 10 Opr/job = 15

Matrik Routing

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	5	3	1	4	2	2	4	5	3	1	3	1	5
2	2	1	4	5	3	1	2	4	5	2	1	3	1
3	5	4	1	2	3	4	5	1	2	1	4	2	2
4	1	5	1	5	1	5	4	2	4	2	3	2	2
5	4	3	2	3	3	2	1	2	2	3	2	3	3
6	3	5	3	5	4	5	5	4	4	4	5	1	2
7	2	3	5	1	2	3	4	2	5	1	3	2	4
8	2	4	5	1	3	4	2	5	2	1	3	4	5
9	2	1	4	5	6	5	3	2	4	2	3	4	5
10	1	2	4	5	3	2	4	5	2	1	2	2	3

	14	15
1	2	3
2	2	5
3	5	1
4	2	4
5	2	3
6	3	1
7	2	4
8	5	1
9	2	3
10	3	2

Kasus 30

Mesin = 10

Job = 5

Opr/job = 20

Matrik Routing

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2	1	2	3	4	5	6	8	7	3	1	2	4
2	10	2	5	4	10	2	10	2	10	1	5	4	9
3	6	2	10	1	2	4	2	3	4	6	7	8	9
4	10	1	2	10	2	4	5	6	1	3	3	5	7
5	1	2	4	5	3	4	6	4	2	4	10	2	3
6	3	10	3	10	1	2	5	4	2	4	5	1	2
7	8	7	8	4	5	7	8	4	5	6	8	4	5
8	5	6	4	10	1	2	4	7	8	6	4	9	7
9	5	10	1	4	5	6	8	9	5	4	2	3	10
10	9	5	6	7	2	4	8	4	10	6	2	2	3

	14	15	16	17	18	19	20
1	2	10	5	4	6	8	6
2	9	5	4	6	1	2	3
3	1	6	4	5	6	10	2
4	4	6	7	6	6	4	3
5	4	5	2	1	10	2	4
6	5	4	5	2	5	2	5
7	1	2	4	5	6	7	8
8	2	10	8	9	7	5	4

9	2	6	1	4	5	6	8
10	5	9	6	7	2	4	8

Matrik Waktu Proses

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	65	56	54	21	20	35	67	92	10	15	13	20	25
2	40	56	23	48	72	56	98	45	51	20	30	16	64
3	30	50	56	31	25	54	56	98	89	89	72	74	56
4	54	61	36	26	54	41	40	50	60	12	36	54	20
5	24	51	23	56	54	21	60	45	80	78	89	87	56
6	64	56	23	21	45	74	84	54	59	68	54	65	43
7	87	54	54	50	58	50	64	24	56	64	54	50	98
8	67	88	87	45	52	64	30	56	89	54	40	57	64
9	50	24	59	65	87	84	54	50	5	4	97	54	55
10	55	66	88	78	55	60	50	50	52	54	55	45	22

	14	15	16	17	18	19	20
1	40	30	64	52	55	48	54
2	50	55	64	66	50	50	30
3	23	24	55	66	60	78	54
4	50	66	33	54	40	25	56
5	58	59	65	42	51	25	21
6	56	24	25	32	24	54	55
7	40	60	45	55	44	55	66
8	60	35	44	58	55	50	88
9	89	88	78	87	90	50	87
10	55	66	80	85	58	88	50

```

USES dos, crt, dasar, start,
input, stime, open, save, chart,
tabel;

```

```

CONST nSiklus = 3;
      nSiklusAkhir = 4;

```

```

PROCEDURE Hyper_Option(no : byte);
  VAR i, posisi : byte;

```

```

BEGIN
  posisi:=3;
  FOR i:=1 TO no-1 DO
    posisi:=posisi+3+length(Menu[i]);
    gotoxy(posisi+1,24);

    textbackground(WarnaTimbulMenuUtama);
    write(Menu[no]);
  END;

```

```

PROCEDURE
Penentuan Waktu_Awal_Per_Job(bantu
M : InfoMesin);
  VAR bantuJ, bantuJT : InfoJob;

```

```

BEGIN
  bantuJ:=bantuM^.NextJ;
  WHILE (bantuJ^.d=0) AND
(bantuJ^.Next<>nil) DO
    bantuJ:=bantuJ^.Next;
    bantuJT:=bantuJ^.Next;
    bantuM^.t:=bantuJ^.r;
    WHILE bantuJT<>nil DO
      BEGIN
        WHILE (bantuJT^.d=0) AND
(bantuJT<>nil) DO

bantuJT:=bantuJT^.Next;

        bantuJ^.t:=bantuM^.t;
        IF ((bantuJT^.r) >
(bantuJ^.t + bantuJ^.d)) AND
(bantuJT<>nil) THEN
          bantuM^.t:=bantuJT^.r
        ELSE bantuM^.t:=bantuJ^.t +
bantuJ^.d;
        bantuJ:=bantuJT;
        IF bantuJT<>nil THEN
bantuJT:=bantuJT^.Next
        END;
        bantuJ^.t:=bantuM^.t;
      END;
    END;

```

```

PROCEDURE
Penentuan_Lintasan(bantuM :
InfoMesin);
  VAR bantuJ, bantuJP : InfoJob;
      cek              : boolean;

BEGIN
  bantuJ:=bantuM^.NextJ;

```

```

    bantuJP:=bantuJ^.Next;
    WHILE bantuJP<>nil DO
      BEGIN
        IF (bantuJP^.d + bantuJP^.q
+ bantuJP^.t) >=
          (bantuJ^.d + bantuJ^.q
+ bantuJ^.t) THEN
          bantuJ:=bantuJP;
          bantuJP:=bantuJP^.Next
        END;
        bantuM^.t:=bantuJ^.d +
bantuJ^.q + bantuJ^.t;

```

```

    bantuJP:=bantuJ;
    cek:=false;
    WHILE bantuJP^.Prev<>nil DO
      BEGIN
        IF (cek=false) AND
(bantuJP^.r=bantuJP^.t) AND
          ((bantuJP^.Prev^.t +
bantuJP^.Prev^.d) < (bantuJP^.t))
        THEN

```

```

          cek:=true;
          bantuJP:=bantuJP^.Prev;
          IF cek=true THEN
            BEGIN
              bantuJP^.d:=0;
            END
          END;
          bantuJP:=bantuJ^.Next;
          WHILE bantuJP<>nil DO
            BEGIN
              bantuJP^.d:=0;
              bantuJP:=bantuJP^.Next
            END
          END;

```

```

PROCEDURE Penentuan_Urutan(bantuM
: InfoMesin);
  VAR bantuJ, bantuJP, temp :
InfoJob;
      cek : boolean;

```

```

BEGIN
  bantuJ:=bantuM^.NextJ;
  WHILE bantuJ^.Next<>nil DO
    BEGIN
      bantuJP:=bantuJ^.Next;
      WHILE (bantuJP<>nil) DO
        BEGIN
          IF (bantuJP<>nil) AND
            ((bantuJP^.r <
bantuJ^.r) OR
              ((bantuJP^.r =
bantuJ^.r) AND (bantuJP^.q >
bantuJ^.q)) OR
              ((bantuJP^.r =
bantuJ^.r) AND (bantuJP^.q =
bantuJ^.q) AND (bantuJP^.d >
bantuJ^.d))) THEN
            BEGIN
              temp:=bantuJP;

```



```

bantuJP:=bantuJP^.Next;

temp^.Prev^.Next:=temp^.Next;
    IF bantuJP<>nil THEN
temp^.Next^.Prev:=temp^.Prev;

        IF
bantuJ^.NoJob=bantuM^.NextJ^.NoJob
THEN
            BEGIN
bantuM^.NextJ:=temp;
                temp^.Prev:=nil
            END
            ELSE
            BEGIN

bantuJ^.Prev^.next:=temp;

temp^.Prev:=bantuJ^.Prev
            END;
            temp^.Next:=bantuJ;
            bantuJ^.Prev:=temp;
            bantuJ:=temp;
        END
        ELSE
        bantuJP:=bantuJP^.Next;
        END;
        bantuJ:=bantuJ^.Next;
        END;
END;

PROCEDURE
Penentuan_Proses_Per_Mesin(bantuM
: InfoMesin);
BEGIN
    Penentuan_Urutan(bantuM);

    Penentuan_Waktu_Awal_Per_Job(bantu
M);
    Penentuan_Lintasan(bantuM)
END;

PROCEDURE Set_Init(no : word; c :
InfoJob);
    VAR temp : InfoInit;
        cek : boolean;

BEGIN
    temp:=Init;
    cek:=false;
    WHILE cek=false DO
        BEGIN
            IF temp^.NextO^.NoJob=no
THEN
                BEGIN
                    temp^.NextO:=c;
                    cek:=true
                END
            ELSE temp:=temp^.Next
            END
        END;
END;

```

```

PROCEDURE MAP;FORWARD;
PROCEDURE Output( bantu :
InfoMesin); forward;

PROCEDURE Cek_Reoptimasi(bantuM :
InfoMesin);
    VAR bantuJ, bantuJP, bantuJT,
        Tcopy, cj, ct :
InfoJob;
        Hcopy, HTcopy :
InfoMesin;
        rMin, dTotal, rTemp :
real;

BEGIN
    bantuJ:=bantuM^.NextJ;
    WHILE (bantuJ^.d=0) AND
(bantuJ<>nil) DO
        bantuJ:=bantuJ^.Next;
        bantuJP:=bantuJ;
        WHILE (bantuJ^.Next^.d<>0) AND
(bantuJ^.Next<>nil) DO
            bantuJ:=bantuJ^.Next;
            bantuJT:=bantuJP;

            WHILE (bantuJT^.q >= bantuJ^.q)
AND (bantuJT <> bantuJ) DO
                bantuJT:=bantuJT^.Next;
            IF bantuJT<>bantuJ THEN
                BEGIN
                    rTemp:=bantuM^.t;
                    cj:=bantuM^.NextJ;

getmem(HTcopy,sizeof(DataMesin));
                    Hcopy:=HTcopy;
                    HTcopy^.NextM:=nil;
                    HTcopy^.NextJ:=nil;
                    HTcopy^.t:=rTemp;

HTcopy^.NoMesin:=bantuM^.NoMesin;
                    WHILE cj<>nil DO
                        BEGIN
getmem(Tcopy,sizeof(DataJob));
                            IF Hcopy^.NextJ=nil THEN
                                BEGIN
                                    Hcopy^.NextJ:=Tcopy;
                                    Tcopy^.Prev:=nil
                                END
                            ELSE
                                BEGIN
                                    ct^.Next:=Tcopy;
                                    Tcopy^.Prev:=ct
                                END;
                            Tcopy^.r:=cj^.r;
                            Tcopy^.d:=cj^.d;
                            Tcopy^.q:=cj^.q;
                            Tcopy^.t:=cj^.t;
                            Tcopy^.NextO:=cj^.NextO;
                            Tcopy^.PrevO:=cj^.PrevO;
                            Tcopy^.NoJob:=cj^.NoJob;

```

```

Tcopy^.Next:=nil;
ct:=Tcopy;
cj:=cj^.Next
END;

IF bantuJP<>bantuJT THEN
rMin:=bantuJP^.r ELSE
rMin:=bantuJP^.Next^.r;
dTotal:=0;
WHILE bantuJP<>bantuJ DO
BEGIN
IF bantuJP<>bantuJT THEN

dTotal:=dTotal+bantuJP^.d;
bantuJP:=bantuJP^.Next;
END;
IF (dTotal+rMin) >
(bantuJT^.r) THEN
BEGIN
bantuJT^.r:=dTotal+rMin;
bantuJ:=bantuM^.NextJ;
END;

Penentuan_Proses_Per_Mesin(bantuM)
;
IF rTemp<=bantuM^.t THEN
BEGIN

bantuM^.NextJ:=Hcopy^.NextJ;
bantuM^.t:=rTemp;
cj:=bantuM^.NextJ;
WHILE cj<>nil DO
BEGIN
bantuJ:=bantuM^.NextJ;
IF cj^.PrevO<>nil THEN
BEGIN
WHILE (bantuJ<>nil)
AND
(cj^.PrevO^.NoJob<>bantuJ^.NoJob)
DO
bantuJ:=bantuJ^.Next;
IF bantuJ<>nil THEN
cj^.PrevO:=bantuJ;
cj^.PrevO^.NextO:=cj
END
ELSE
Set_Init(cj^.NoJob,cj);

bantuJ:=bantuM^.NextJ;
IF cj^.NextO<>nil THEN
BEGIN
WHILE (bantuJ<>nil)
AND
(cj^.NextO^.NoJob<>bantuJ^.NoJob)
DO
bantuJ:=bantuJ^.Next;
IF bantuJ<>nil THEN
cj^.NextO:=bantuJ;
cj^.NextO^.PrevO:=cj;
END;

```

```

cj:=cj^.Next
END
END;
END;

PROCEDURE Isi_tMesin(bantu :
InfoMesin);
VAR bantuM : InfoMesin;
bantuJ : InfoJob;

BEGIN
bantuM:=bantu;
WHILE bantuM<>nil DO
BEGIN
bantuJ:=bantuM^.NextJ;
WHILE bantuJ<>nil DO
BEGIN
IF bantuJ^.d<>0 THEN
bantuM^.t:=bantuJ^.d+bantuJ^.t;
bantuJ:=bantuJ^.Next;
END;
bantuM:=bantuM^.NextM
END
END;

FUNCTION Cari_Bottleneck(bantuM :
InfoMesin): InfoMesin;
VAR bantuMP : InfoMesin;
bantuJ : InfoJob;
cek : boolean;
i, j : word;

BEGIN

bantuJ:=bantuM^.NextJ;
i:=0 ;
WHILE bantuJ<>nil DO
BEGIN
IF bantuJ^.d<>0 THEN i:=i+1;
bantuJ:=bantuJ^.Next;
END;

bantuMP:=bantuM^.NextM;
WHILE bantuMP<>nil DO
BEGIN
cek:=false;
IF bantuMP^.t=bantuM^.t THEN
BEGIN
j:=0;
bantuJ:=bantuMP^.NextJ;
WHILE bantuJ<>nil DO
BEGIN
IF bantuJ^.d<>0 THEN
j:=j+1;
bantuJ:=bantuJ^.Next;
END;
IF j>i THEN cek:=true;
END;
IF (bantuMP^.t>bantuM^.t) OR
(cek=true) THEN

```

```

BEGIN
    i:=j;
    bantuM:=bantuMP;
END;
bantuMP:=bantuMP^.nextM;
END;
Cari_Bottleneck:=bantuM;
END;

PROCEDURE Ubah_r(bantu : InfoJob);
VAR pOpr : InfoJob;
    temp : real;

BEGIN
    temp:=bantu^.t + bantu^.d;
    pOpr:=bantu^.NextO;
    WHILE pOpr<>nil DO
    BEGIN
        IF pOpr^.r<temp THEN
        BEGIN
            pOpr^.r:=temp;
            temp:=pOpr^.r + pOpr^.d;
            pOpr:=pOpr^.NextO;
        END
        ELSE pOpr:=nil;
    END;
END;

PROCEDURE Tetapkan_d;
VAR tInit : InfoInit;
    tJob : InfoJob;
    i,j : word;

BEGIN
    tInit:=Init;
    i:=1;
    WHILE tInit<>nil DO
    BEGIN
        tJob:=tInit^.NextO;
        j:=1;
        WHILE tJob<>nil DO
        BEGIN
            tJob^.d:=Get_tProses(i,j);
            j:=j+1;
            tJob:=tJob^.NextO
        END;
        i:=i+1;
        tInit:=tInit^.Next
    END
END;

PROCEDURE Nilai_Penyebaran(bantu :
InfoMesin);
VAR bantuJ, pOpr : InfoJob;
    temp : real;

BEGIN
    Tetapkan_d;
    bantuJ:=bantu^.NextJ;
    Ubah_r(bantuJ);

```

```

WHILE bantuJ^.Next<>nil DO
BEGIN
    Ubah_r(bantuJ^.Next);
    bantuJ:=bantuJ^.Next;
END;
temp:=bantuJ^.d+bantuJ^.q;
bantuJ:=bantuJ^.Prev;
WHILE bantuJ<>nil DO
BEGIN
    IF bantuJ^.q<temp THEN
    BEGIN
        bantuJ^.q:=temp;
        pOpr:=bantuJ^.PrevO;

temp:=bantuJ^.q+bantuJ^.d;
        WHILE pOpr<>nil DO
        BEGIN
            IF pOpr^.q<temp
THEN
                BEGIN
                    pOpr^.q:=temp;

temp:=temp+pOpr^.d;
                    pOpr:=pOpr^.PrevO;
                END
                ELSE pOpr:=nil;
            END;
            temp:=bantuJ^.q +
bantuJ^.d;
            bantuJ:=bantuJ^.Prev;
        END;
    END;

PROCEDURE Putuskan_Rantai(VAR
bantu, Intf : InfoMesin);
VAR temp : InfoMesin;

BEGIN
    IF bantu^.NextM<>nil THEN
    bantu^.NextM^.PrevM:=bantu^.PrevM;
    IF bantu^.PrevM=nil THEN
    Intf:=bantu^.NextM
    ELSE
    bantu^.PrevM^.NextM:=bantu^.NextM;
    bantu^.PrevM:=nil;
    IF AfterScj=nil THEN
    AfterScj:=bantu
    ELSE
    BEGIN
        temp:=AfterScj;
        WHILE temp^.NextM<>nil DO
            temp:=temp^.NextM;
        bantu^.PrevM:=temp;
        temp^.NextM:=bantu;
    END;
    bantu^.NextM:=nil;
END;

PROCEDURE Masuk_AfterScj(bantu :
InfoMesin);
BEGIN

```

```

        Nilai_Penyebaran(bantu);
Putuskan_Rantai(bantu,BeforeScj);
END;

PROCEDURE Proses_Inti(Node :
InfoMesin);
    VAR bantuM : InfoMesin;
BEGIN
    bantuM:=Node;
    WHILE bantuM<>nil DO
        BEGIN
            IF Node=AfterScj THEN
                BEGIN
                    Tetapkan_d;
                END;
        Penentuan_Proses_Per_Mesin(bantuM)
        ;
            Cek_Reoptimasi(bantuM);
            IF Node=AfterScj THEN
                Nilai_Penyebaran(bantuM);
                bantuM:=bantuM^.NextM
            END;
        END;

PROCEDURE Urutkan_AfterScj;
    VAR bantuM, temp, temp1 :
InfoMesin;
BEGIN
    bantuM:=AfterScj;
    WHILE bantuM^.NextM<>nil DO
        BEGIN
            temp:=bantuM^.NextM;
            WHILE temp<>nil DO
                BEGIN
                    IF
temp^.NoMesin<bantuM^.NoMesin THEN
                        BEGIN
                            temp1:=temp;
temp^.PrevM^.NextM:=temp^.NextM;
                            IF temp^.NextM<>nil
THEN
                                temp^.NextM^.PrevM:=temp^.PrevM;
                                temp:=temp1^.NextM;
                                IF bantuM=AfterScj
THEN AfterScj:=temp1
                                    ELSE
bantuM^.PrevM^.NextM:=temp1;
                                    temp1^.NextM:=bantuM;
                                    IF bantuM=AfterScj
THEN temp1^.PrevM:=nil
                                        ELSE
temp1^.PrevM:=bantuM^.PrevM;
                                        bantuM^.PrevM:=temp1;
                                        bantuM:=temp1;
                                    END
                                ELSE temp:=temp^.NextM;
                                END;
                            END;
                        END;

```

```

        bantuM:=bantuM^.NextM
    END
END;

PROCEDURE Rangking;
    VAR hTemp, bantu, temp :
InfoMesin;
BEGIN
    Isi_tMesin(AfterScj);
    hTemp:=AfterScj;
    AfterScj:=nil;
    WHILE hTemp<>nil DO
        BEGIN
            bantu:=Cari_Bottleneck(hTemp);
            Putuskan_Rantai(bantu,hTemp)
        END;
    END;

PROCEDURE Proses;
    VAR bantuM : InfoMesin;
        i, iterke : byte;
        tS, tF : real;
BEGIN
    afterscj:=nil;
    iterke:=0;

    gettime(Jam, Menit, Detik, Seper100);
    tS:=Detik+Seper100/100;
    WHILE BeforeScj<>nil DO
        BEGIN
            iterke:=iterke+1;
            Proses_Inti(BeforeScj);

            bantuM:=cari_Bottleneck(BeforeScj)
            ;
                IF iterke=1 THEN
                    LB:=bantuM^.t;
                    Masuk_AfterScj(bantuM);
                    FOR i:=1 TO nSiklus DO
                        BEGIN
                            Proses_Inti(AfterScj);
                            Rangking;
                        END;
                        bantuM:=BeforeScj;
                    END;
                    FOR i:=1 TO nSiklusAkhir DO
                        BEGIN
                            Proses_Inti(AfterScj);
                            Rangking;
                        END;
                        bantuM:=BeforeScj;
                    Urutkan_AfterScj;

                    gettime(Jam, Menit, Detik, Seper100);
                    tF:=Detik+Seper100/100;
                    Waktu:=tF-tS;
                    goal:=3
                END;

```

```

PROCEDURE Map;
  VAR tInit : InfoInit;
      tJob  : InfoJob;

BEGIN
  Hapus_Area;
  textattr:=black*16+lightgray;
  clrscr;
  tInit:=Init;
  WHILE tInit<>nil DO
  BEGIN
    tJob:=tInit^.NextO;
    WHILE tJob<>nil DO
    BEGIN
      write(tJob^.nojob, '
',tJob^.r:2:0, ' ');
      write(tJob^.d:2:0, '
',tJob^.q:2:0, ' ',tJob^.t:2:0);
      write(' ');
      tJob:=tJob^.NextO;
    END;
    writeln; writeln;
    tInit:=tInit^.Next;
  END;
  READLN;
END;

```

```

PROCEDURE Output( bantu :
InfoMesin);
  VAR bantuM : InfoMesin;
      bantuJ : InfoJob;

BEGIN
  textattr:=black*16+lightgray;
  clrscr;
  bantuM:=bantu;
  WHILE bantuM<>nil DO
  BEGIN
    writeln('Mesin
',bantuM^.NoMesin); writeln;
    bantuJ:=bantuM^.NextJ;
    WHILE bantuJ<>nil DO
    BEGIN
      write(bantuJ^.NoJob, '
',bantuJ^.r:2:0);
      writeln('
',bantuJ^.d:2:0, '
',bantuJ^.q:2:0, '
',bantuJ^.t:2:0);
      bantuJ:=bantuJ^.Next;
    END;
    readln;
    bantuM:=bantuM^.NextM;
  END;
  readln;
END;

```

```

PROCEDURE Ubah_Pilihan(VAR no :
byte ; mode : boolean);

```

```

  VAR i, posisi : byte;

BEGIN
  posisi:=3;
  FOR i:=1 TO no-1 DO
  posisi:=posisi+length(Menu[i])+3;

  textbackground(WarnaDasarMenuUtama
);
  gotoxy(posisi+1,24);
  write(Menu[no]);

  textbackground(WarnaTimbulMenuUtama
a);
  IF mode=false THEN IF no=1 THEN
no:=9 ELSE no:=no-1
  ELSE IF no=9 THEN no:=1 ELSE
no:=no+1;
  posisi:=3;
  FOR i:=1 TO no-1 DO
  posisi:=posisi+length(Menu[i])+3;
  gotoxy(posisi+1,24);
  write(Menu[no])
END;

```

```

PROCEDURE To Menu Chart;
  VAR tombol, tmb : char;

```

```

BEGIN
  IF goal<>0 THEN
  BEGIN
    Show Menu Utama(true);
    Tampilan_Awal_Chart;
    REPEAT
      tombol:=readkey;
      tombol:=upcase(tombol);
    UNTIL (tombol=#27) OR
(tombol='M') OR (tombol='O');

    textattr:=WarnaDasarArea*16+yellow
;
    IF tombol<>#27 THEN
    BEGIN
      IF (goal=1) OR (goal=4)
THEN Proses;
      Init_Graph;
      REPEAT
        Hapus_Area_Grafik;
        tmb:=tombol;
        CASE tombol OF
          'M' :
            tombol:=Hasil_M;
          'J' :
            tombol:=Hasil_O;
        END;
        IF (tombol<>#27) AND
(tombol<>'M') AND (tombol<>'O')
THEN
          BEGIN
            REPEAT
              tombol:=readkey;

```

```

tombol:=upcase(tombol)
      UNTIL
(((tombol='M') OR (tombol='O'))
AND (tombol<>tmb))
      OR
(tombol=#27)
      END
      UNTIL tombol=#27;
      Close_Graph;
      Tampilan_Utama;
      Hyper_Option(6);
    END;
  END
  ELSE
Error(1,WarnaDasarArea,WarnaTeksMe
nuUtama);
      Hapus_Area
    END;

PROCEDURE To_Menu_Tabel;
BEGIN
  IF goal<>0 THEN
    BEGIN
      IF (goal=1) OR (goal=4) THEN
        Proses;
        Gambar_Tabel;

      END
    ELSE
      Error(1,WarnaDasarArea,WarnaTeksMe
nuUtama);
      Hapus_Area
    END;

PROCEDURE Keluar;
BEGIN
  release(tPointer);
  textattr:=black*16+lightgray;
  clrscr;
  writeln('Terima Kasih Anda
Telah Memakai Software Ini Untuk
Penyelesaian Job Shop');
  writeln;
  Cursor(6,7);
  halt
END;

PROCEDURE Pilih_Menu(no : byte);
BEGIN
  CASE no OF
    1 : To_Menu_Start;
    2 : To_Menu_Input;
    3 : To_Menu_FirstTime;
    4 : To_Menu_Open;
    5 : To_Menu_Save;
    6 : To_Menu_Chart;
    7 : To_Menu_Tabel;

```

```

      8 : ;
      9 : Keluar
    END;
    Hyper_Option(no)
  END;

PROCEDURE Option;
  VAR Pilihan : byte;
      Tombol : char;

BEGIN
  clrscr;
  Tampilan_Utama;
  goal:=0;
  dOpr:=OprMax+1; dJob:=JobMax+1;
  dMesin:=MesinMax+1;
  Hyper_Option(1);
  Pilihan:=1;
  REPEAT
    Tombol:=readkey;
    IF Tombol=#0 THEN
      BEGIN
        Tombol:=readkey;
        CASE Tombol OF
          #75 :
            Ubah_Pilihan(Pilihan, false);
          #77 :
            Ubah_Pilihan(Pilihan, true);
        END
      END
    ELSE
      IF Tombol=#13 THEN
        Pilih_Menu(Pilihan)
        UNTIL Tombol=#27; { ESC }
        Keluar;
      END;

PROCEDURE Read_File_Ini;
  TYPE tipeisi = string[20];
  VAR teks : text;
      letak : byte;
      judul : string[50];
      isi : tipeisi;
      strtmp : string[70];

  FUNCTION Warna(strColor :
tipeisi) : byte;
    VAR i : byte;
    CONST pC : ARRAY[1..16]
OF tipeisi =
('BLACK','BLUE','GREEN','CYAN','RE
D','MAGENTA','BROWN',

'LIGHTGRAY','DARKGRAY','LIGHTBLUE'
,'LIGHTGREEN',

'LIGHTCYAN','LIGHTRED','LIGHTMAGEN
TA','YELLOW','WHITE');

```

```

BEGIN
  FOR i:=1 TO
length(strColor) DO
    strColor[i]:=upcase(strColor[i]);

    i:=0;
    REPEAT
      i:=i+1
    UNTIL strColor=pC[i];
    Warna:=i-1
  END;

```

```

BEGIN
  assign(teks,'ta.ini');
  reset(teks);
  WHILE NOT eof(teks) DO
    BEGIN
      readln(teks,strttmp);
      letak:=pos('=',strttmp);
      IF (pos('[',strttmp)=0) AND
(letak>0) THEN
        BEGIN
          isi:=copy(strttmp,letak+1,length(st
rtmp)-letak);
          letak:=pos(' ',isi);
          WHILE letak>0 DO
            BEGIN
              delete(isi,letak,1);
              letak:=pos(' ',isi)
            END;
          letak:=pos(' ',strttmp);

```

```

judul:=copy(strttmp,1,letak-1);

  IF judul='GraphDirectory'
THEN GraphDirectory:=isi
  ELSE IF judul='WarnaLogo'
THEN WarnaLogo:=Warna(isi)
  ELSE IF
judul='WarnaDasarArea' THEN
WarnaDasarArea:=Warna(isi)
  ELSE IF
judul='WarnaDasarBingkaiArea' THEN
WarnaDasarBingkaiArea:=Warna(isi)
  ELSE IF
judul='WarnaBayanganBingkai' THEN
WarnaBayanganBingkai:=Warna(isi)

```

```

  ELSE IF
judul='WarnaDasarMenuUtama' THEN
WarnaDasarMenuUtama:=Warna(isi)
  ELSE IF
judul='WarnaTeksMenuUtama' THEN
WarnaTeksMenuUtama:=Warna(isi)
  ELSE IF
judul='WarnaTimbulMenuUtama' THEN
WarnaTimbulMenuUtama:=Warna(isi)

```

```

  ELSE IF
judul='WarnaTeksTampilanStart'
THEN

```

```

WarnaTeksTampilanStart:=Warna(isi)
  ELSE IF
judul='WarnaTimbulTampilanStart'
THEN
WarnaTimbulTampilanStart:=Warna(isi)
  ELSE IF
judul='WarnaDasarMenuStart' THEN
WarnaDasarMenuStart:=Warna(isi)
  ELSE IF
judul='WarnaTeksPokokMenuStart'
THEN
WarnaTeksPokokMenuStart:=Warna(isi)
  ELSE IF
judul='WarnaTeksPenjelasMenuStart'
THEN
WarnaTeksPenjelasMenuStart:=Warna(
isi)
  ELSE IF
judul='WarnaDasarMenuInput' THEN
WarnaDasarMenuInput:=Warna(isi)
  ELSE IF
judul='WarnaTeksPokokMenuInput'
THEN
WarnaTeksPokokMenuInput:=Warna(isi)
  ELSE IF
judul='WarnaTeksPenjelasMenuInput'
THEN
WarnaTeksPenjelasMenuInput:=Warna(
isi)
  ELSE IF
judul='WarnaBingkaiTampilanInput'
THEN
WarnaBingkaiTampilanInput:=Warna(i
si)
  ELSE IF
judul='WarnaTeksTetapTampilanInput'
THEN
WarnaTeksTetapTampilanInput:=Warna
(isi)
  ELSE IF
judul='WarnaTeksIsianTampilanInput'
THEN
WarnaTeksIsianTampilanInput:=Warna
(isi)
  ELSE IF
judul='WarnaPenjelasIsianInput'
THEN
WarnaPenjelasIsianInput:=Warna(isi)
  ELSE IF
judul='WarnaDasarMenuFirstTime'
THEN
WarnaDasarMenuFirstTime:=Warna(isi)
  ELSE IF
judul='WarnaTeksPokokMenuFirstTime'
THEN
WarnaTeksPokokMenuFirstTime:=Warna
(isi)

```

```

        ELSE IF
judul='WarnaTeksPenjelasMenuFirstTime' THEN
WarnaTeksPenjelasMenuFirstTime:=Warna(isi)
        ELSE IF
judul='WarnaBingkaiTampilanFirstTime' THEN
WarnaBingkaiTampilanFirstTime:=Warna(isi)
        ELSE IF
judul='WarnaTeksTetapTampilanFirstTime' THEN
WarnaTeksTetapTampilanFirstTime:=Warna(isi)
        ELSE IF
judul='WarnaTeksIsianTampilanFirstTime' THEN
WarnaTeksIsianTampilanFirstTime:=Warna(isi)
        ELSE IF
judul='WarnaPenjelasIsianFirstTime' THEN
WarnaPenjelasIsianFirstTime:=Warna(isi)

        ELSE IF
judul='WarnaTeksAwalTampilanChart' THEN
WarnaTeksAwalTampilanChart:=Warna(isi)

        ELSE IF
judul='WarnaTeksTabel' THEN
WarnaTeksTabel:=Warna(isi)
        ELSE IF
judul='WarnaBingkaiTabel' THEN
WarnaBingkaiTabel:=Warna(isi)

        ELSE IF
judul='WarnaDasarError' THEN
WarnaDasarError:=Warna(isi)
        ELSE IF
judul='WarnaTeksError' THEN
WarnaTeksError:=Warna(isi)
        END;
    END
END;

```

```

{ Program Utama }
BEGIN
    Read_File_Ini;
    mark(tPointer);
    Is_Save:=false;
    NamaFile:='';
    Option;
END.

```